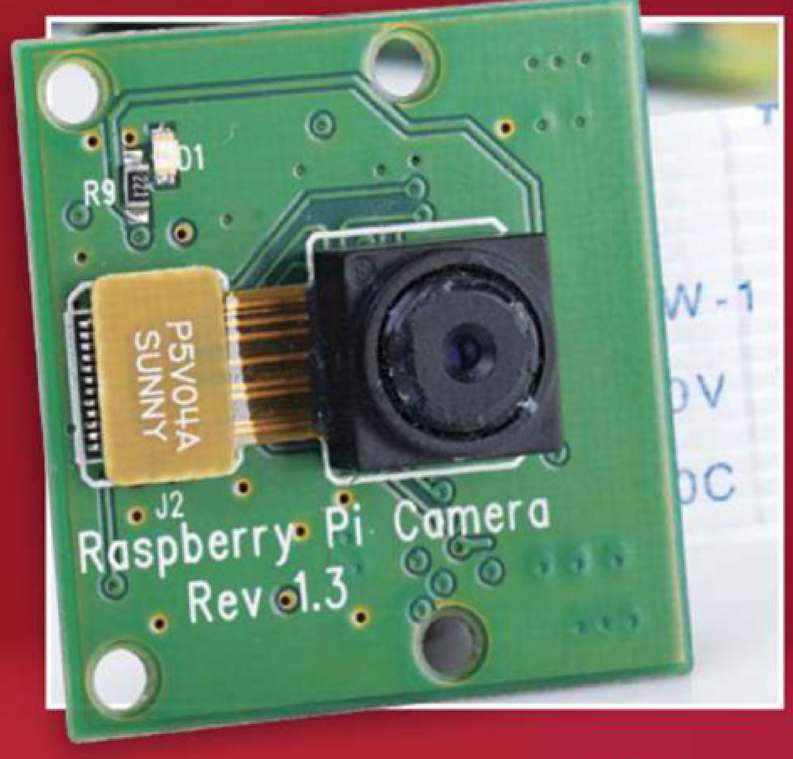
RasPi

DESIGN
BUILD
CODE

10

Get hands-on with your Raspberry Pi







10 INSPIRING PROJECTS

Arcade, time-lapse video, weather station **& more**

Welcome



Right then, it's project time this month! With summer just around the corner and us Pi people itching to try out some new ideas, we thought we'd round up some excellent hardware and software projects for you, all ready for a few weekends of hacking and making. We've got ten for you to try out, covering everything from a mini arcade cabinet through to a working weather station – and best of all, you can even combine some of these projects. Why not add some home-made dusklights to your weather station so that they automatically switch on at night? We've also powered-up Talking Pi this month with a massive round-up of the most frequently asked Pi questions – hope it helps!

Gavin Thomas

Deputy Editor

Get inspired

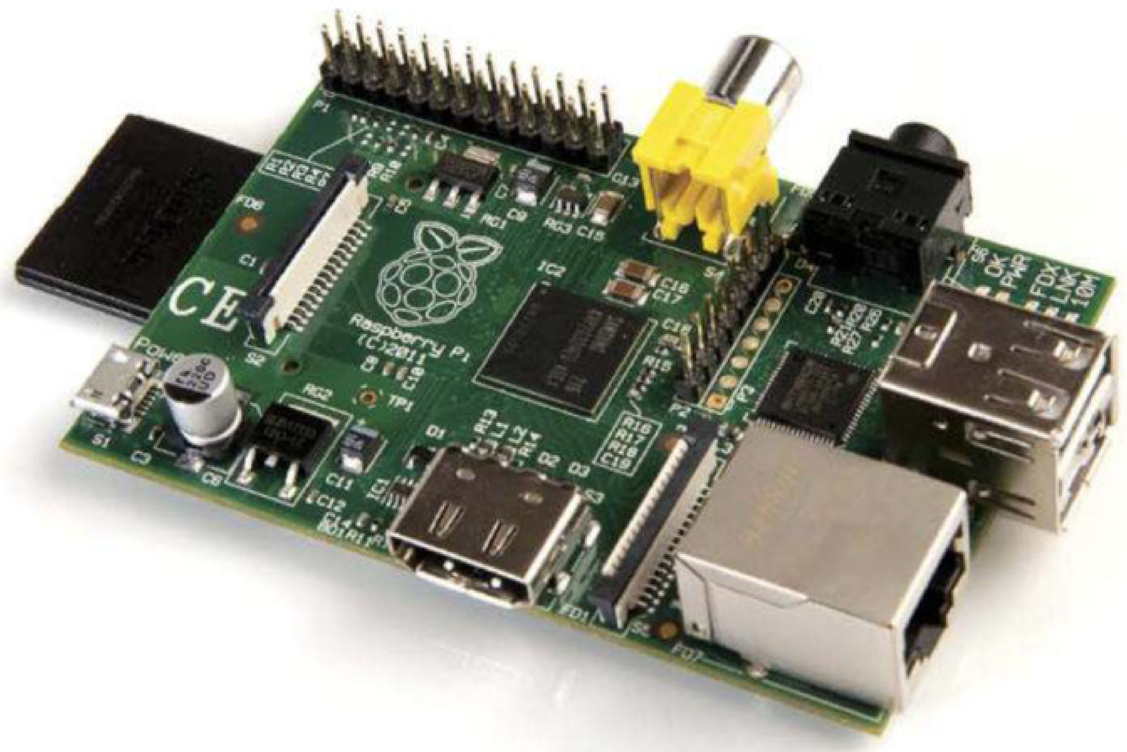
Discover the RasPi community's best projects

Expert advice

Got a question? Get in touch and we'll give you a hand

Easy-to-follow guides

Learn to make and code gadgets with Raspberry Pi



From the makers of

LinuxUser & Developer

Join the conversation at...

@linuxusermag | Linux User & Developer | RasPi@imagine-publishing.co.uk



Contents

Retro arcade cabinet
Run MAME on Adafruit's Cupcade

Audiobook reader
Play books and podcasts with one button

Web radio
Stream online radio stations on the go

Media caster
Broadcast music and movies on your network

Portable Wi-Fi signal repeater
Bad connection upstairs? Use this repeater

Secure Tor web station
Browse the Internet in complete privacy

Private cloud storage
Use your Raspberry Pi as a personal Dropbox

AirPi
Predict the weather with your own sensors

Dusklights
Light up your garden without touching a switch

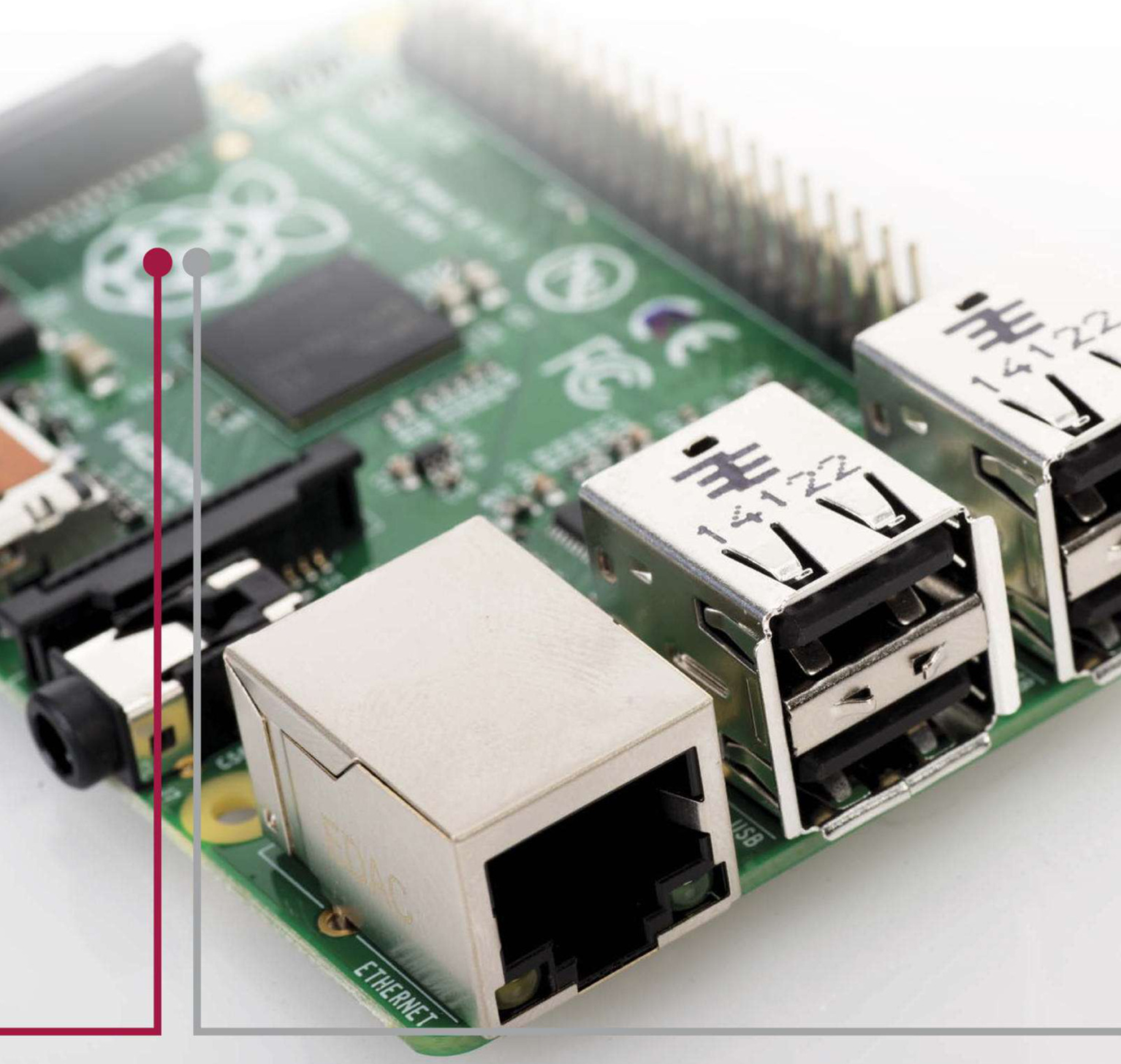
Outdoor time-lapse camera
Capture every second of a sunset

Talking Pi – Bonus Round
The ultimate Raspberry Pi FAQ



10 Inspiring Pi projects

Fire up your imagination and test your Python skills



The Raspberry Pi has come a long way over the last couple of years, and it's all down to the creativity and imagination of a burgeoning community that's bought over five million of these single-board computers. They've powered quadcopters and coffee makers, self-sailing boats and even touched the edge of space, but the one thing that's still as much in demand as it was on launch day is an idea.

Our Raspberry Pi community works so well because we all share our work with each other. We can all see what everyone else is up to and, if we see something that we like, we can have a go at making it ourselves – and then changing it, so we can share a different version. It's all about inspiration.

So for this issue of **RasPi** we made ten inspiring projects for you to try – and we encourage you to deviate from the steps when a new idea takes hold and let us know where your project leads you! There's a range of different projects here for you to try, covering media and entertainment, networks, security and more. All the components you'll need for each one are listed on the page, and if you swipe to the end of the projects then you'll find a further resources round-up. You can also get the full code for these projects from our website.

One more thing to mention – some of our Raspberry Pi creations are designed to be combined, to give you a great starting point when you begin adapting these projects to the way you want them. If you come up with any combinations we've missed, be sure to tweet us a photo – and have fun making them!



Below This is the Cupcade – the tiniest arcade cabinet you can get for a Pi



Retro arcade cabinet

Create a tiny arcade machine and learn how to expand it into a fully-stocked games cabinet

Our office is home to the legendary Retro Gamer magazine, and when the Raspberry Pi was released a couple of years ago the team immediately recognised its potential as an excellent device to power a homemade, MAME-using arcade machine. After a chat between our editor and Retro Gamer's, the team was satisfied that maybe one day it would use a Raspberry Pi or another Linux-powered machine to run a cabinet in the future.

They weren't the only ones that realised how well a Pi would fit in an arcade machine; many projects have been completed since then with impressive results. If you don't

THE PROJECT ESSENTIALS

Adafruit Cupcade
adafruit.com/product/1783s

1.5 A power supply

Soldering iron

Wire strippers

Short length of audio wire



Below You can reverse the button mapping for classic NES-style control

fancy carving out custom pieces of wood just yet then you can always start small with the Cupcade kit from Adafruit. The kit is composed of laser-cut plastic panels, a PiTFT screen and a selection of buttons, custom boards and a special joystick that can be assembled into a miniature MAME arcade cabinet.

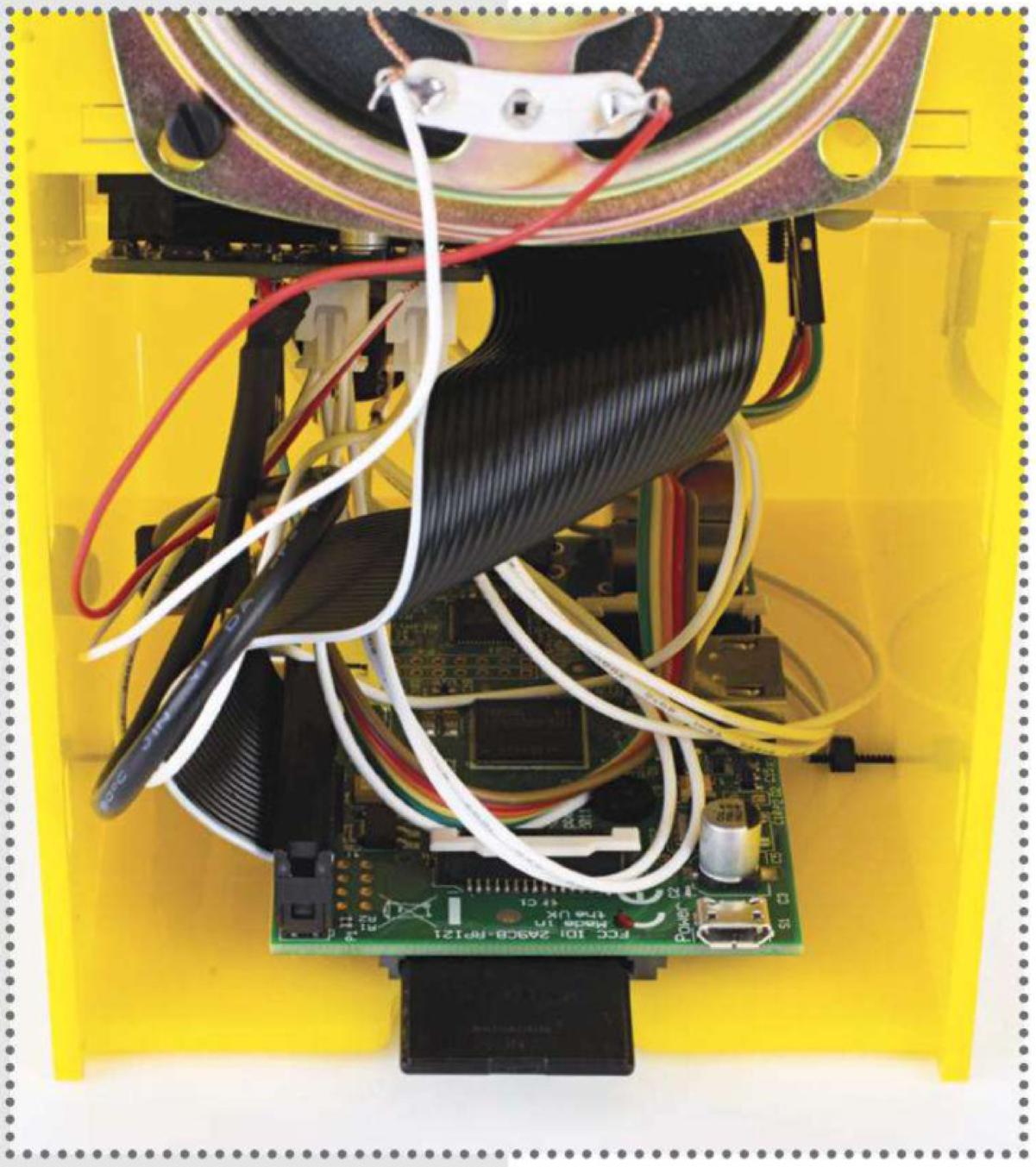
Unfortunately, for the moment, the Cupcade only supports the original Model B. However, we like to think of it as a great excuse to test out whether or not your Model B is up to the task of powering an arcade machine in case you want to expand. If not, that means you'll need another Raspberry Pi to replace the arcade one, in which case you now have the perfect reason to head out and grab yourself a new Raspberry Pi 2.

When you get the kit in, you'll need to assemble the components step-by-step. It's a long process which we don't quite have room for in this issue, so instead we're going to point you towards the excellent Adafruit instructions for constructing your Cupcade: **https://learn.adafruit.com/cupcade-raspberry-pi-micro-mini-arcade-game-cabinet**.

A brief word of warning: it will take you a few hours to complete, so make sure you have some time clear to do so. It's quite fiddly in places as well, so it might be worth getting a second pair of hands to help you.

When attaching the side panel to the screen, speaker and buttons, we suggest ignoring the instruction to carefully lift the panel off your work desk. Instead, try sliding it slightly off the desk itself so that you can access

Below The Raspberry Pi sits snugly below a great bale of wires



each screw one at a time without having to precariously balance the sections at an angle.

Once completed, it's time to test it out. Adafruit has a pre-rolled image you can write to an SD card which can be downloaded with:

```
$ wget http://adafruit-download.s3.amazonaws.
com/cupcade-5-13-2014.img.zip
```

Write it to a 4 GB card (your Cupcade comes with one), download the free, non-commercial ROMs they provide to test the setup and power it up with a proper plug adapter. The components are running directly off the Pi and can be quite power-hungry, so it's imperative that you properly power it.

The image we've written has a full MAME arcade emulator in it, and a NES emulator. They're both configured to use the buttons you've installed on the case by default; this means that because the NES emulator directly maps the cabinet's A and B buttons to its own A and B, the layout is the reverse of the NES controller, which had A on the right and B on the left. Depending on what you plan to use the cabinet for, you can either swap over the buttons or just get used to a slightly altered control scheme.

How can we expand the Raspberry Pi from here to create a full-sized cabinet? The good news is that the emulators will still work when you use the HDMI port for output instead, allowing you to hook it up to a normal TV. The buttons and speakers will still work as well and they can be plugged straight into the Raspberry Pi's



Be inspired

One of the best arcade cabinet projects we've seen was featured on the Raspberry Pi blog: **bit.ly/1pWjQWE**. Using an old arcade cabinet for parts, the maker modified the inputs to be read by the Pi and got the coin slot to work. You don't need to go this far, but it certainly adds a flair of authenticity to a project like this.

GPIO port via a connecting ribbon, like we used with the board to the screen. However, this does leave you with only four buttons and the small joystick; luckily there are plenty of USB arcade sticks and components that can be connected via USB, and with the right configuring you can get them working on the emulators just fine.

There's a lot more you can do with the Pi once you've got it set up in a cabinet, such as letting it connect to the network so you can remotely maintain it without having to disassemble the entire thing to retrieve an SD card. Make sure you do your research on specific parts and look out for any community forums that will definitely have helpful advice on what to buy.



> "The emulators will still work when you use the HDMI port to hook it up to a normal TV"



Audiobook reader

Create an audiobook and podcast player that works at the touch of a button

Audiobooks and podcasts are becoming more and more prevalent in a society where we've learned to multi-task thanks to all our modern devices. Whether out and about or doing some chores, it can be nice to have some extra sensory input when you're not using your full concentration. Enter, like always, the Raspberry Pi. With only a small selection of components and a pair of earphones or portable speakers, you can take your Raspberry Pi with you and have it play audiobooks.

First, you need to get the Raspberry Pi ready. Do your usual **apt-get update** and **apt-get upgrade** in the terminal before you start, although there isn't really much preparation involved. Once that's done, all we need to do is install VLC, which is done with:

```
$ sudo apt-get install vlc
```

Once that's done, turn off your Pi, disconnect the power and begin hooking up your circuit according to the Fritzing diagram on the next page. It's a fairly simple setup – we're going to use the button to activate the playing of the audiobook and the LED will turn on while this is happening.

THE PROJECT ESSENTIALS

Speakers
LED
1.2k resistor
10k resistor
Push-button switch
Breadboard and wires

> "We're going to use the button to activate the playing of the audiobook and the LED will turn on while this is happening"

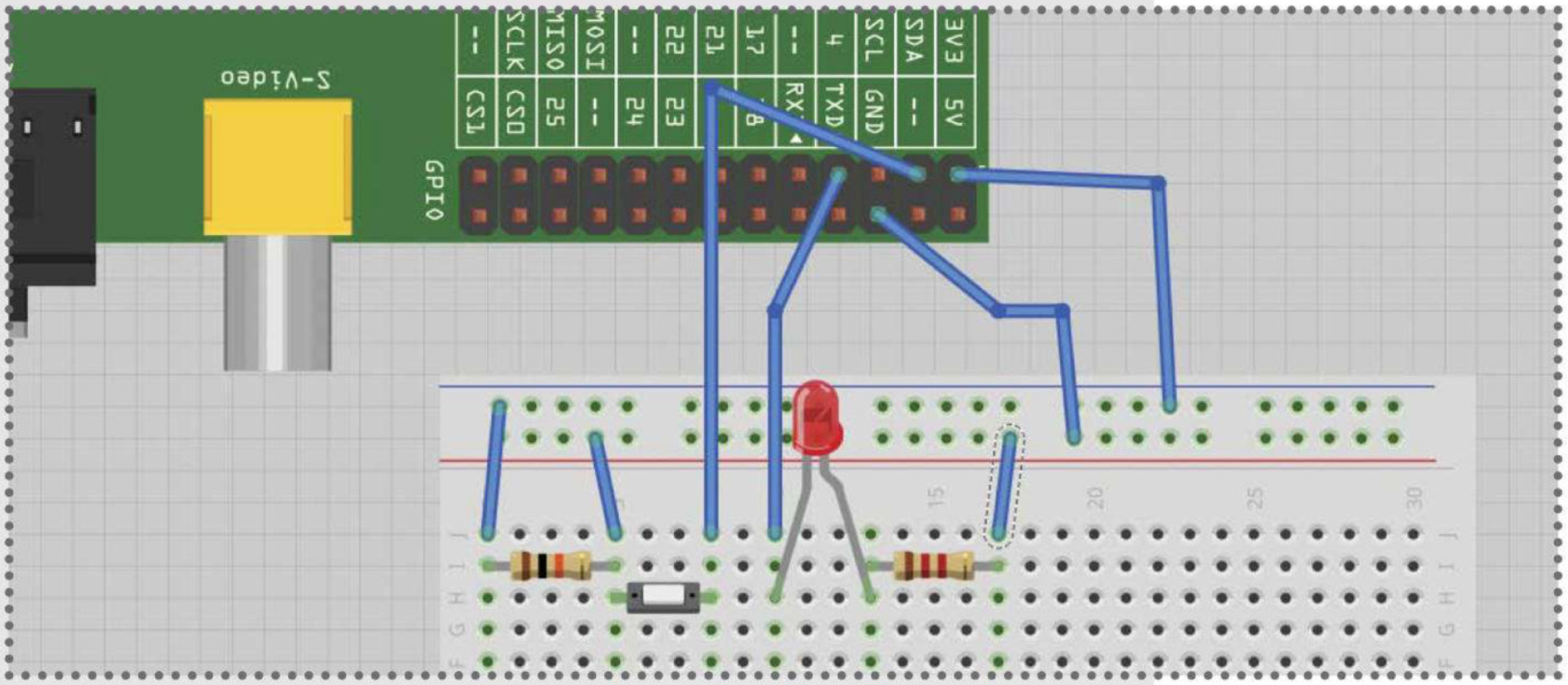
Once you're confident with the circuit, turn your Raspberry Pi back on and open up the terminal. Type in the following to download the Python code for this project:

```
$ wget http://www.linuxuser.co.uk/wp-content/
uploads/2014/08/audiobook.zip
```

There are a couple things we need to note about the code: it first of all assumes you have your audiobook located in a directory known as audiobook within your home folder. It also looks for a file called audiobook.mp3 to play; you can easily change this by editing the path but you'll need to be consistent with the name of the file you have on your Pi. You could always have it point to a specific location on a USB stick, swapping out the files on another computer and renaming them appropriately once you've finished with previous book.

Test it out by running the code – one press of the button will start the process while another press will stop playing the book. If you find the button press too sensitive or not sensitive enough, edit the delay in time.sleep which is currently set at a default of 0.2 seconds.

Below This circuit will work on the Model B, Model B+ and the Raspberry Pi 2 – the pins begin at the same end of each board





Web radio

A portable radio that streams web content wherever you are

Our Raspberry Pi web radio is a true companion piece to the audiobook reader we just looked at, and we'll get into why as we go through this tutorial. Like the audiobook, we're taking full advantage of the Raspberry Pi to help aid you in a real-life situation. As standard radio methods begin to die out, the established web-streaming platforms will become more and more popular, and you can get in on those straight away using this project. The circuit for this is also the exact same one as the audiobook, so if you've already had a crack at that, then you'll just need to grab the code for this tutorial.

THE PROJECT ESSENTIALS

Speakers
LED
1.2k resistor
10k resistor
Push-button switch
Breadboard and wires



Below The synergy between this and the audiobook reader means they can share the same setup

Before you wire it up, it's time to prepare your Raspberry Pi. Open the terminal and do an **apt-get update** and **apt-get upgrade** to make sure everything is up to date, followed up by installing VLC using:

```
$ sudo apt-get install vlc
```

When it's finished installing, you will need to turn off your Raspberry Pi. Disconnect the USB cable, break out the breadboard and components and follow the wiring diagram from the audiobook reader guide. It's not a complicated circuit at all, and revolves around sensing a button being pressed and lighting an LED on command. Double-check all your connections on the breadboard and then plug your Pi back in. Open the terminal and use it to download our Python script for playing internet radio:

```
$ wget http://www.linuxuser.co.uk/wp-content/
uploads/2014/08/webradio.zip
```

Unzip the file and put it wherever you want; it doesn't require any special placement. You will likely want to change the radio station that the code is currently tuned to, and this can be found in the subprocess.Popen line in the code. Find an m3u URL for your favourite radio station and then replace the URL already in that code to get it to play.

There may be a short delay between pressing the button and the music starting as VLC launches and gets the stream buffered. We've made it so the light comes on straight away, though, so at least you'll know the process has started. Pressing the button again will kill VLC and stop the radio stream. If you feel the button is too sensitive or not sensitive enough, find the time.delay setting and change the amount of time it requires the button to be pressed.

Put 'em together

You can easily put both projects together and have an all-in-one radio and audiobook player. Wire up a second button that will play the audiobook instead and then maybe even add extra buttons for different radio stations.



Media caster

Cast your content over the local network to receive it from multiple clients so you don't lose your place

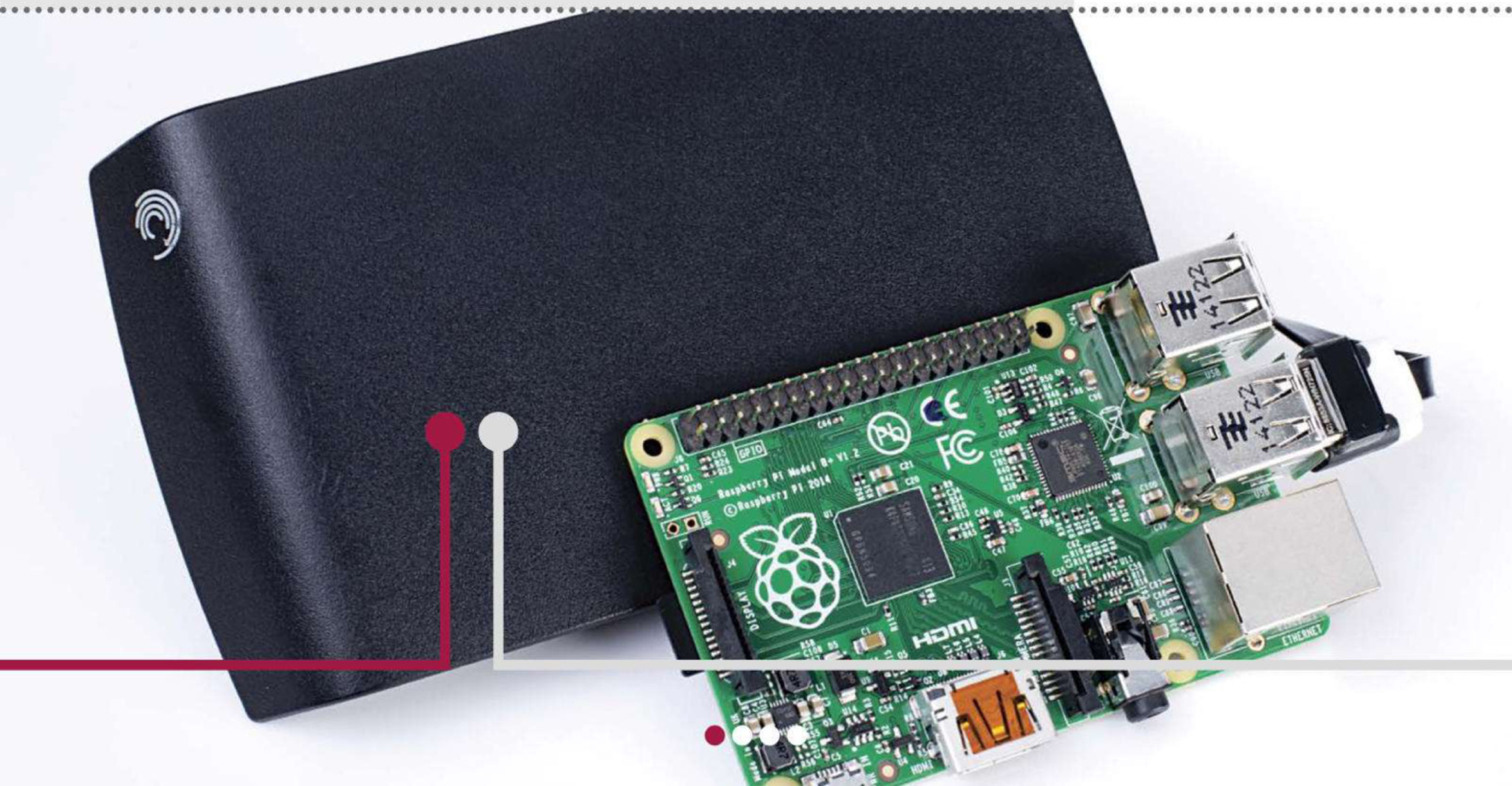
Have you ever wanted to move between rooms while watching or listening to the same film or song? We often find ourselves switching from a phone or tablet to the desktop. There are some complicated setups you can use for this involving MythTV and various custom-built servers and receivers, but you can also do it with just one Raspberry Pi server and then many Kodi receivers with a lot less hassle.

Broadcasting your media over your network so that the clients can pick up the stream means that you can easily pick up where you left off when switching devices, without having to note down the track position and re-find the file.

THE PROJECT ESSENTIALS

Kodi
kodi.tv/

Portable hard drive



Below Use a nice, big external storage device to hold all of your music and videos

01 Install storage drivers

You'll likely be using a portable NTFS hard drive if you want to store a lot of media on your Raspberry Pi caster. This means you need to install the NTFS libraries on the Pi, which requires you to open the terminal and type:

```
$ sudo apt-get install ntfs-3g
```

02 Create mounting folder

We're going to have the Raspberry Pi automatically mount the hard drive for us whenever it boots up, and in order to do this it first needs to have a place where the files will be accessible. Let's create a directory called mediadrive inside the Home folder. Keep the directory name to just one word to make your life easier later.

```
$ mkdir mediadrive
```

03 Boot-time mount parameters

Your hard drive will likely be mounted as /dev/sda1, but do a **fdisk -l** if you want to double check. To make sure it mounts at boot, go to fstab with **sudo nano /etc/fstab** and write the following line below, with a tab between each bit of information:

```
/dev/sda1      /home/pi/mediadrive      ntfs-3g
defaults,noatime      0      0
```

04 Install VLC on Pi

So all our media is now accessible, and now we need a way to broadcast it. For this we'll use multi-functional

"We're going to have the Raspberry Pi automatically mount the hard drive for us whenever it boots up"

storemags.com

media player VLC, which was originally designed to be the client for a casting server. You can install it with:

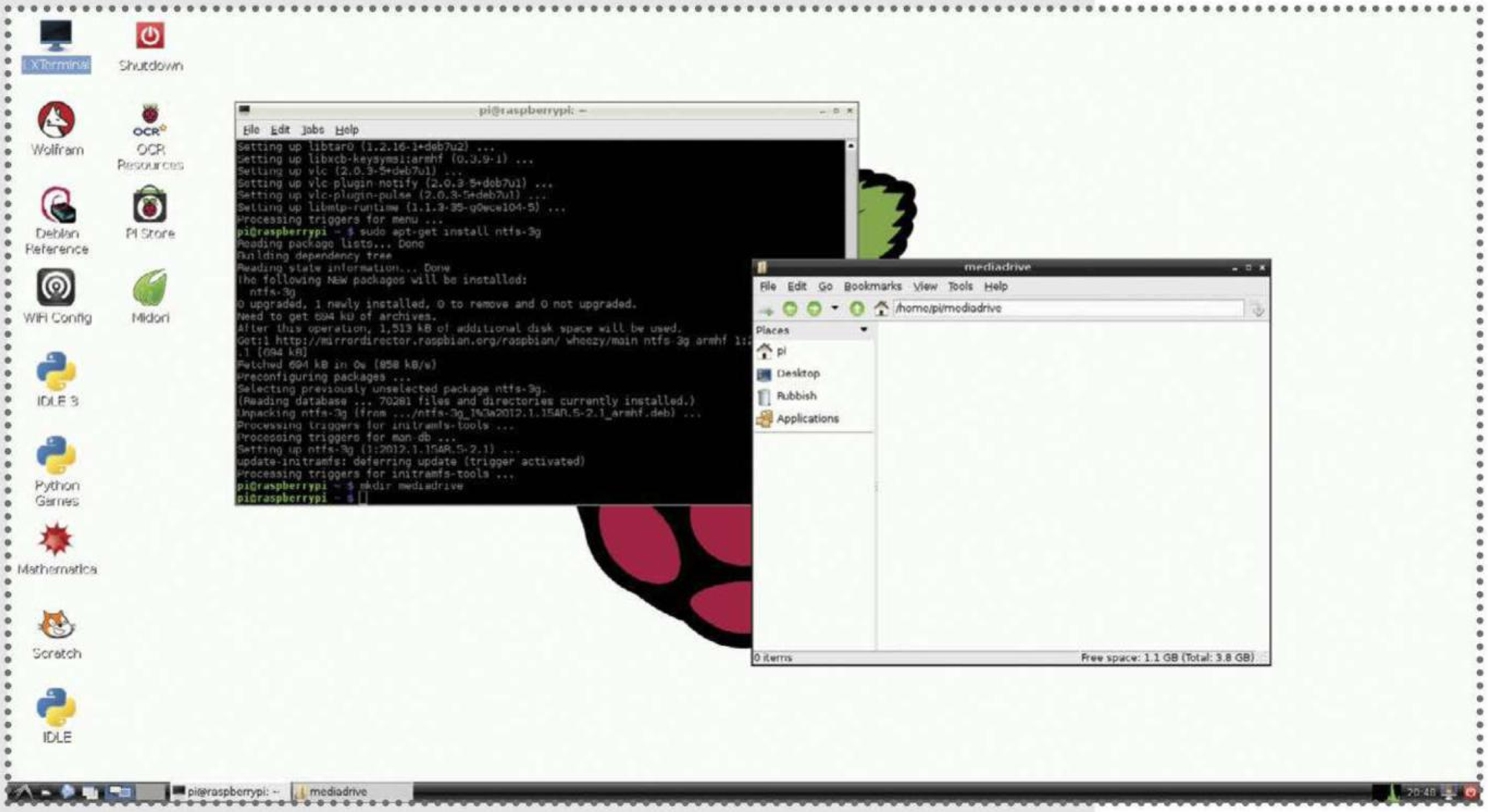
```
$ sudo apt-get install vlc
```

05 Kodi stream receiver

This bit is dead simple – on any computer you can easily get files off of, create a file called Stream.strm, and add this line to it, with IP as the last two sets of Pi's IP address:

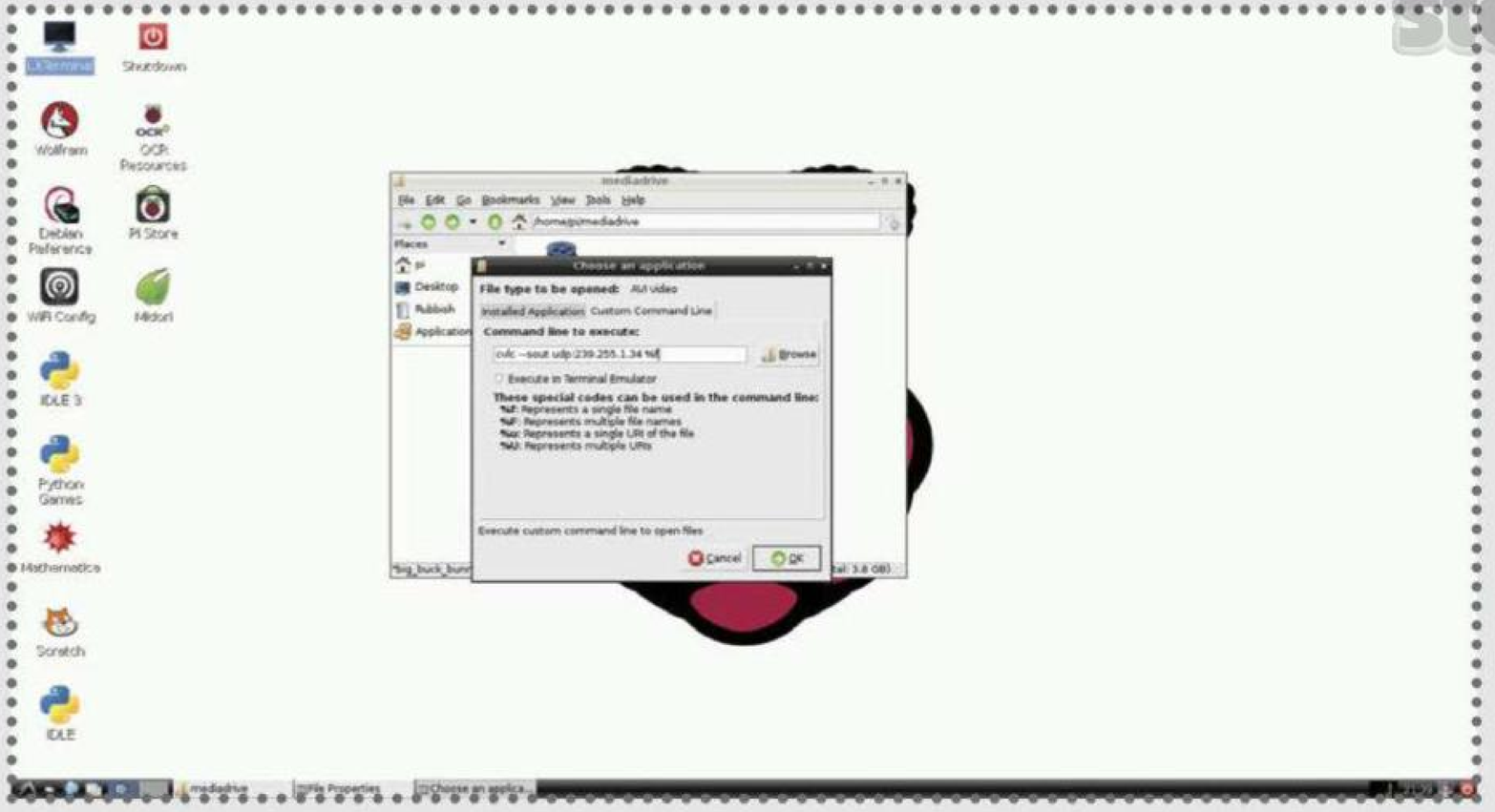
```
udp://239.255.[IP].[IP]:1234
```

Save it and move it to your Kodi clients that you want to receive the cast from.



Below Media stored on your portable hard drive will show up in a dedicated folder

"SSH into your Pi from another computer or an Android device"



06 Custom VLC command

Here's the really fun part. Go back to your Pi and right-click on one of the files you'd like to play, go to Properties and click the down arrow next to Open with. Go to Customise and then Custom Command Line and type in the following, with IP as the last two sets on your IP address:

```
cvlc --sout udp:239.255.[IP].[IP] %f
```

07 Test out casting

Here's the big test: boot up Kodi, open your media on the Raspberry Pi and use the Stream file on Kodi to receive the cast. It might take a moment to load but it will start playing via Kodi in short time.

08 Command line alternative

If it's not convenient for you to launching files from the Pi desktop, you can always SSH into your Pi from another computer or an Android device then run the file using a similar custom command to the one we used earlier:

```
$ cvlc --sout udp:239.255.[IP].[IP] [file
location]
```

Be inspired

Why not add a web interface to your Pi so that you can browse from your laptop or smartphone? Have it run the multicast code when clicking on files it's scraped from the media drive to make the process much easier. You can also fuse the project with our cloud server in a few pages time.

storemags



Storemags.com

Wi-Fi signal repeater

Boost your signal using a Raspberry Pi and two USB Wi-Fi dongles

What we like about the Pi for this project is that it's very low maintenance and extremely easy to put in an appropriate location. All you need is two USB wireless adaptors, a nice little case and a way to power it. Once set up, you can leave it to its own devices, checking in over SSH every now and then to do updates.

Set this up on a monitor with a keyboard, using a fresh image of Raspbian and with your Wi-Fi dongles plugged in. On first boot keep it as CLI and after doing the usual **apt-get update/upgrade**, type **startx** to get to the desktop. Configure the wireless for wlan0 to connect to your home network and make sure it has a fixed IP address, then

THE PROJECT ESSENTIALS

2 USB Wi-Fi adaptors
Raspberry Pi case
Portable power pack



Below One Wi-Fi adaptor picks up your main Wi-Fi network, while the other one then broadcasts a new signal

reboot to make sure it all works from the command line by using **ping www.google.com**. Now you need to install your first bit of software using:

```
$ sudo apt-get install hostapd iw
```

After it's installed, download and save the file we've created directly to config by entering the following two commands:

```
$ wget http://www.linuxuser.co.uk/wp-content/
uploads/2014/08/repeater.zip
$ cp LUDRepeater.conf /etc/hostapd/hostapd.conf
```

Do a reboot and test the configuration file with:

```
$ hostapd -dd /etc/hostapd/hostapd.conf
```

If there are no errors, open the file using **nano /etc/hostapd/hostapd.conf** and then add the following to it:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
RUN_DAEMON=yes
```

Also change the SSID to be the same one for your network in the ssid field. Next, install the bridge utilities with **apt-get install bridge-utils**. Now configure it with these commands:

```
brctl addbr bridge0
brctl addif bridge0 wlan0
brctl addif bridge0 wlan1
ifconfig bridge0 up
```

Test it out to make sure it works and then place it around the house to extend your wireless signal!

"Change the SSID to be the same one for your network in the ssid field. Once that's done, you'll need to install the bridge utilities"



Secure Tor web station

Stay private online by routing all your web traffic through Tor on your Raspberry Pi

In a much more privacy-focused world, being able to browse securely online is an important freedom for many people. With the use of Tor and a few tweaks to the Raspberry Pi, you can make sure all your Internet traffic is kept private.

First of all, you'll need to make sure to install Tor from the repos. Open up the LXTerminal and simply type:

```
$ sudo apt-get install tor
```

THE PROJECT ESSENTIALS

2 USB Wi-Fi adaptors
Raspberry Pi case
Portable power pack



Below Adafruit's Onion Pi Pack contains all the components you need to set up a secure connection

Once that's done, edit the torrc file by using **sudo nano /etc/tor/torrc** and add this to the top of the file:

```
VirtualAddrNetworkIPv4 10.192.0.0/10
AutomapHostsOnResolve 1
TransPort 9040
DNSPort 53
```

Save this and then open the next file with **sudo nano /etc/resolv.conf** and modify it:

```
nameserver 127.0.0.1
```

Finally, you also need to change the iptables ruleset, but before you do this, use **top** to confirm the uid of Tor and make a note of it. Now open up a new file with **nano /etc/init.d/iptables** and enter the code found at: www.linuxuser.co.uk/wp-content/uploads/2014/08/tor.zip. Now save it and then enter:

```
$ chmod 755 /etc/init.d/iptables
$ update-rc.d iptables defaults 12
```



Put 'em together

A secure wireless connection can be created using Onion Pi: **https://learn.adafruit.com/onion-pi/overview**. This sets up a wireless access point that anonymises web traffic by directing it through the Tor network, so all your devices connect to Tor and not just the Pi itself. Apply the lessons learned in the Wi-Fi Repeater tutorial to unshackle it from the ethernet cable, although that will slightly reduce its overall security.

Left You can also find the secure Tor Browser and Tails distro on the Tor Project website

storemags.com



Private cloud storage

Turn the Pi into your own personal Dropbox using ownCloud and an internet connection

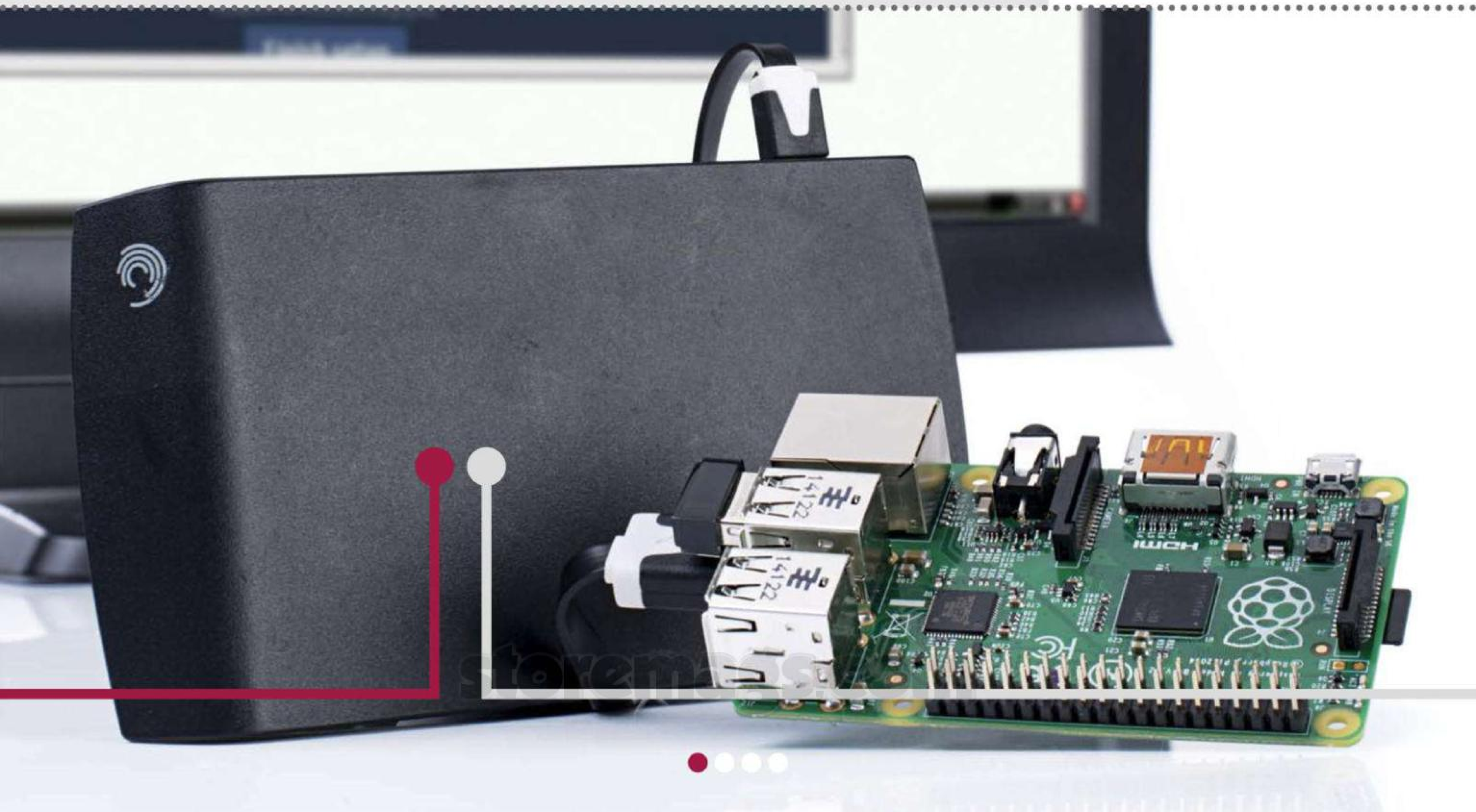
The themes of a lot of our Raspberry Pi guides revolve around the size and portability of the Pi itself, lending it to tasks you may have used a full-sized or small computer for in the past that the Pi can now take over. Having your own private cloud is another excellent use of the Raspberry Pi's capabilities, because you can store it hidden away somewhere and it will require very little day-to-day maintenance.

Make sure you invest in some decent, portable USB storage such as an external HDD, and also get a case for your Pi in the process.

THE PROJECT ESSENTIALS

Portable hard drive

Below Storing your files locally is far more secure than keeping them on a third-party server



01 Set a static IP

After setting up your wired or wireless internet connection, you need to make it static. Use **sudo nano /etc/network/interfaces** to open up the network settings file. Find the iface eth0 line so you can change and add this to it:

```
iface eth0 inet static
    address 192.168.0.50
    gateway 192.168.0.1
    netmask 255.255.255.0
    network 192.168.0.0
    broadcast 192.168.0.255
```

02 Install a lot of software

You'll want Apache software and PHP for this. Install everything you need with:

```
$ sudo apt-get install apache2 php5 php5-json
php5-gd php5-sqlite curl libcurl3 libcurl4-
openssl-dev php5- curl php5-gd php5-cgi php-
pear php5-dev build-essential libpcre3-dev
libapache2-mod-php5 php-apc
```

03 Set up PHP accelerator

Install your accelerator with **sudo pecl install apc** and create an ini file for it. To do this, use **sudo nano /etc/php5/cgi/conf.d/apc.ini** and then add this to the file:

```
extension=apc.so
apc.enabled=1
apc.shm_size=30
```

04 Configure file limits

Go into the Apache config file with **sudo nano /etc/php5/apache2/php.ini**. It's a big file, but there are two

"After setting up your wired or wireless internet connection, you need to make it static"

> "You need to enable SSL in Apache; do this by changing 'None' to 'All' inside AllowOverride"

filesize options you need to find, and a third extension option you need to add as below:

```
upload_max_filesize = 2048M
post_max_size = 2200M
extension = apc.so
```

05 Set up SSL

First of all, you need to enable SSL in Apache; do this by using **sudo nano /etc/apache2/sites-enabled/000-default** and changing 'None' to 'All' inside the AllowOverride option. Follow this up with the following two commands:

```
$ sudo a2enmod rewrite
$ sudo a2enmod headers
```

06 Finish up with Apache

You now need to do two sets of commands: a big one that requires some info and then a restart:

```
$ sudo openssl genrsa -des3 -out server.key 1024
$ sudo openssl rsa -in server.key -out server.
key.insecure
$ sudo openssl req -new -key server.key -out
server.csr
$ sudo openssl x509 -req -days 365 -in server.
csr -signkey server.key -out server.crt
$ sudo cp server.crt /etc/ssl/certs;sudo cp
server.key /etc/ssl/private;sudo a2enmod ssl
$ sudo a2ensite default-ssl
$ sudo service apache2 restart
```

07 Download and install ownCloud

Here you have a series of commands to run that will download, unzip and then install ownCloud to the required place:

```
$ wget https://download.owncloud.org/community/owncloud-7.0.0.tar.bz2
$ sudo tar -xjf owncloud-7.0.0.tar.bz2
$ sudo cp -r owncloud /var/www
```

08 ownCloud permissions

Make sure your hard drive is connected and mounted as you want it. First, you need to give webserver permission to use ownCloud:

```
$ sudo chown -R www-data:www-data /var/www/owncloud/
```

Next, use **sudo nano /var/www/owncloud/.htaccess** and make the same changes you did in the php.ini file earlier. Finally, give permissions to the location you mounted the hard drive to with:

```
$ sudo chown -R www-data:www-data [mount]
```

09 Set up ownCloud

Open Midori and navigate to https://[ipaddress]/owncloud to begin the ownCloud setup process. The first thing you'll need to do is change the data location to the mount point of your external drive, which can be found in the advanced options. And you're done!

Be inspired

Need an idea of what your next step could be? Check out our media caster tutorial and maybe you can fuse the two concepts together. Upload videos to your cloud and get them to play over your network.

Below You'll see a warning in ownCloud if you haven't got everything set up nice and securely





Storemags

AirPi

Predict the weather using your own sensor station

The AirPi has a pedigree that aligns perfectly with the mission of the Raspberry Pi Foundation. Created by two teens in sixth form to measure various forms of pollution, the project's popularity exploded with thanks to some highlighting by the Raspberry Pi Foundation and it's now the premier way to create a weather station on the Raspberry Pi.

The AirPi kit comes with a small selection of sensors by default that can be upgraded and improved upon with extra breakout boards and modules. Included with the kit is a barometric pressure sensor, a humidity and temperature

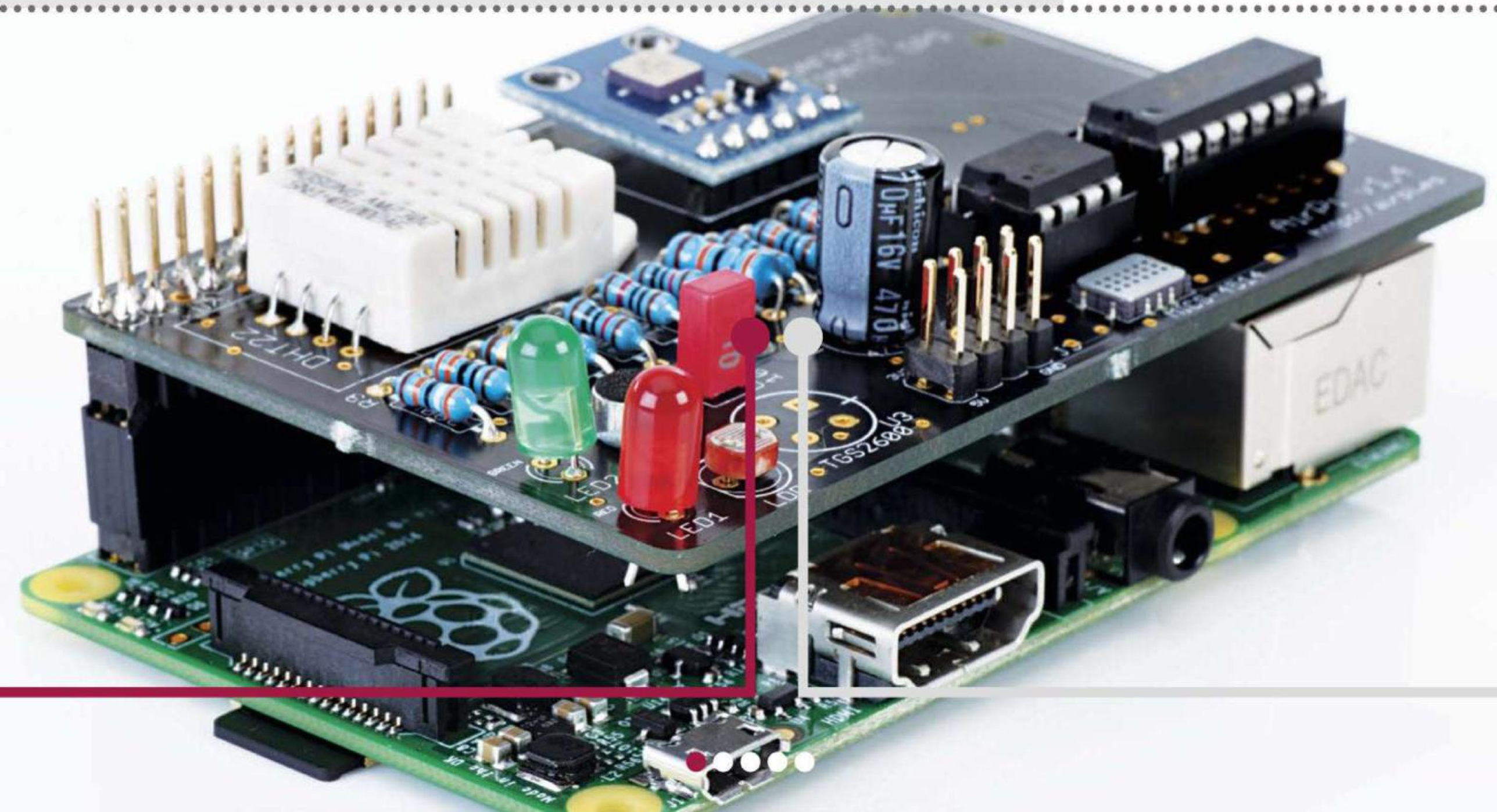
THE PROJECT ESSENTIALS

AirPi

Soldering iron

Solder

Below The AirPi perfectly fits the Raspberry Pi, so you can easily enclose your device in a small weatherproof box



sensor, an ultraviolet radiation sensor and a microphone for noise pollution.

The first thing you'll need to do is assemble the kit, for which you'll require a soldering iron and a steady hand. There are some basic instructions that come with the kit for you to follow, but it doesn't hurt to have a look at some of the online images to figure out exactly where everything goes. Be careful when soldering all the resistors to make sure that you don't melt the 26-pin connector on the underside, otherwise it won't go on your Pi.

Once that's done you need to start setting up your Pi. Open a terminal to get some of the Python tools you'll need:

```
$ sudo apt-get install git-core python-dev
python-pip python-smbus
```

Download and install the python-eeml package:

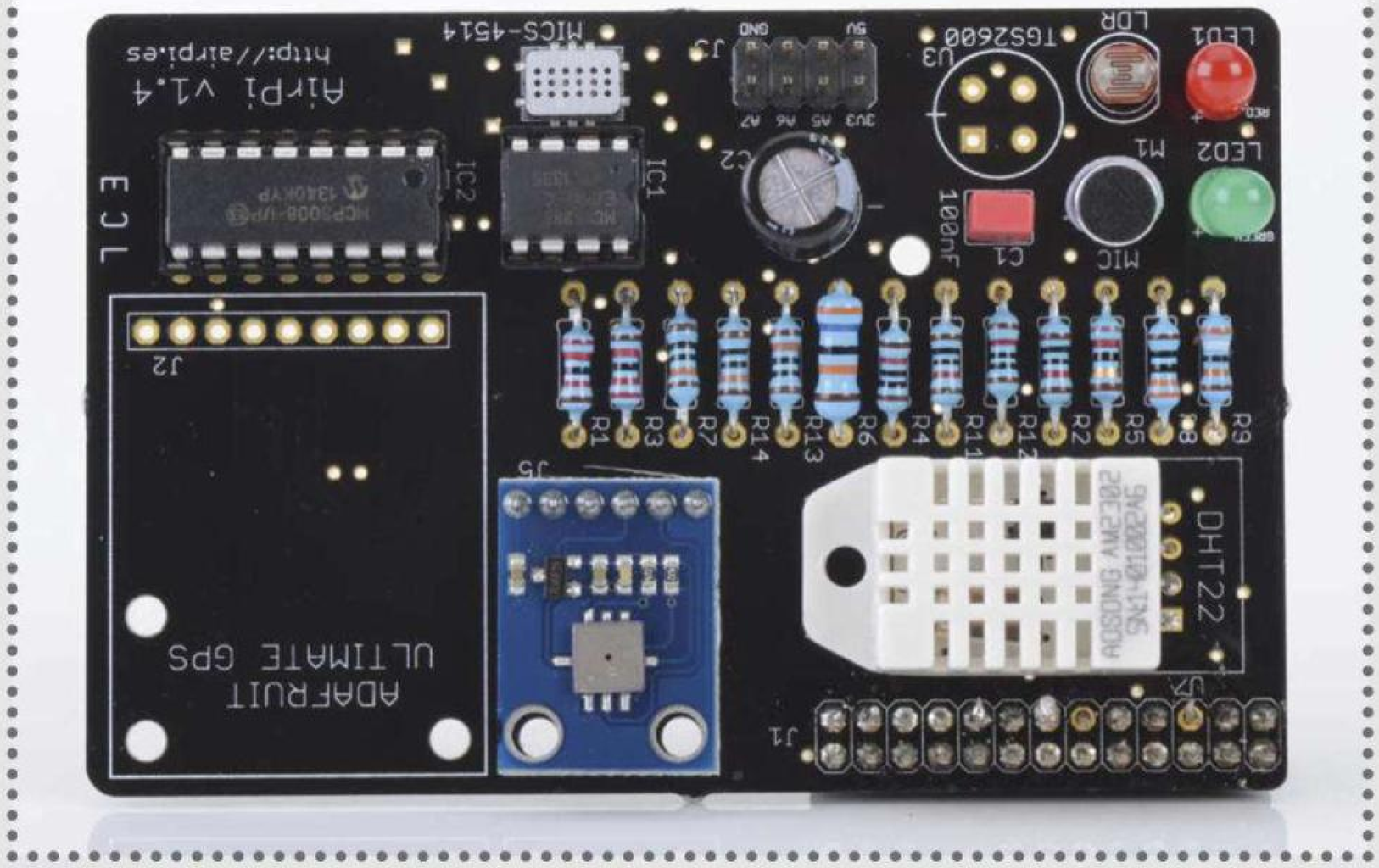
```
$ sudo apt-get install libxml2-dev libxslt1-
dev python-lxml
$ git clone https://github.com/petervizi/
python-eeml.git
$ cd python-eeml
$ sudo python setup.py install
```

You need to install i2c support for the pressure sensor, so first use **sudo nano /etc/modprobe.d/raspi-blacklist.conf** and remove the comment from the **blacklist i2c-bcm2708** line. Next, open up the modules file with **sudo nano /etc/modules** and add **i2c-dev** to the bottom of the file. Save that and install:

```
$ sudo apt-get install i2c-tools
```

"Included is a barometric pressure sensor, a humidity and temperature sensor, an ultraviolet radiation sensor and a microphone for noise pollution"

StoreMags.com



Left Measure your surroundings with AirPi and share your data with the scientific community

Add your Pi to the i2c user group with:

```
$ sudo adduser pi i2c
```

Now reboot your Raspberry Pi before continuing. Finally, install the final Python module with:

```
$ sudo apt-get install python-smbus
```

Now we can install the AirPi code. Still in the terminal, use:

```
$ git clone -b non-modular https://github.com/
tomhartley/AirPi.git
```

If you're using a revision 2 Model B or a B+, use **sudo nano AirPi/AirPi.cfg** to edit the config file. Change **I2CBus = 0** to **I2CBus = 1** and alter the next line to read **LCD = False**. Create a Xively account, add the API and ID keys to the config file and save it. You can now start taking measurements by running Upload.py.

"If you're using a revision 2 Model B or a B+, edit the config file. Change I2CBus = 0 to I2CBus = 1 and alter the next line to read LCD = False"



Dusklights

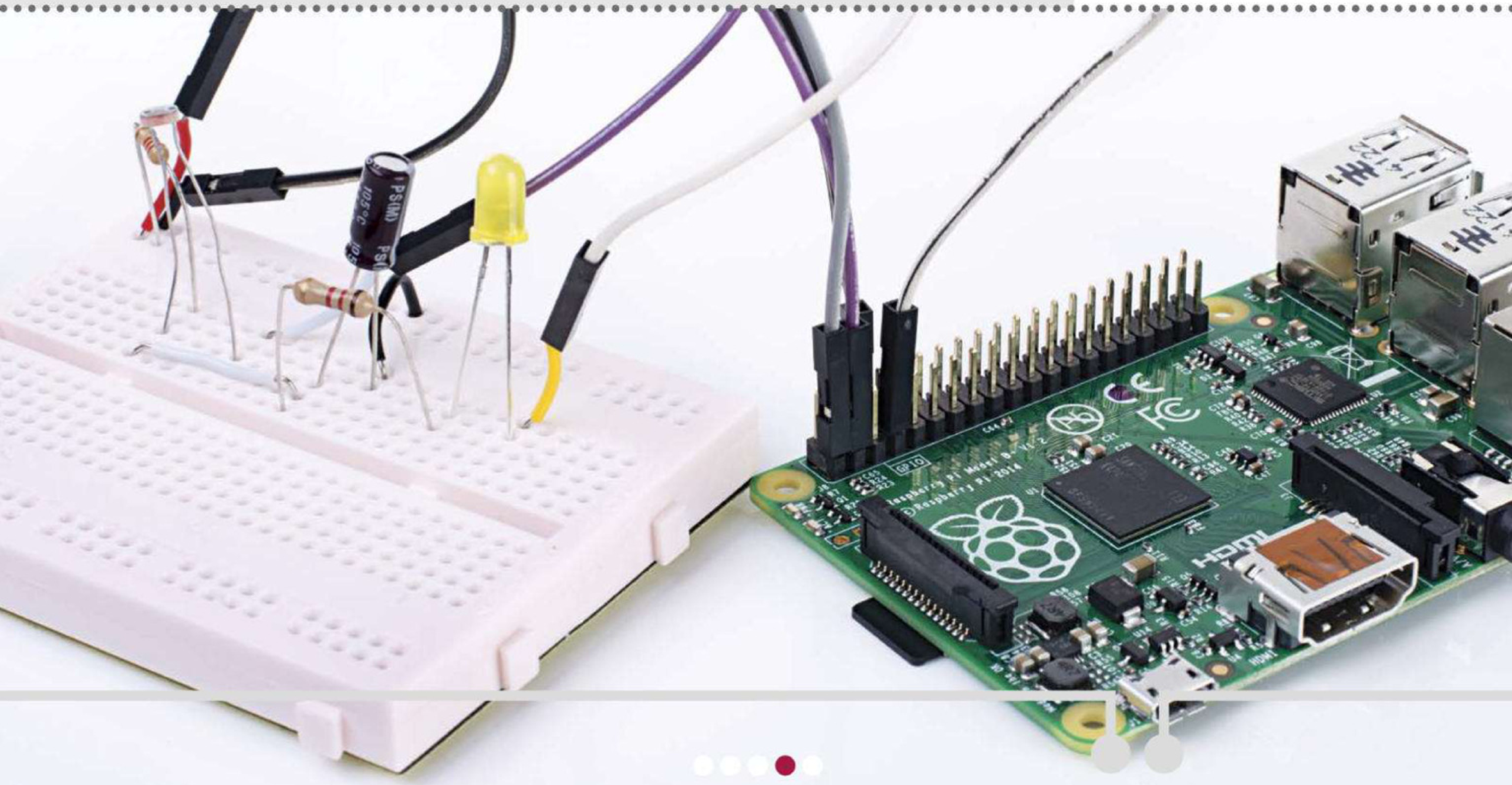
Use your Pi to control outdoor lights that automatically come on when it's dark

You've probably all seen those cheap solar-powered lamps that you can stick into your garden to try and give it a classy bit of illumination during the night. If you've actually got one then you may have found out that they don't shine very brightly, and the plastic stakes can be very flimsy. So why not make your own version? What we'll show you on this page is the beginning of an array of light-sensitive LEDs using a single LED, so that you can understand how the system works. We'll use a special resistor called a light dependant resistor

THE PROJECT ESSENTIALS

1.2k resistor
2.2k resistor
Light dependent resistor
1µF capacitor
Breadboard and wires

Below Our small example can easily be scaled up using the same principles

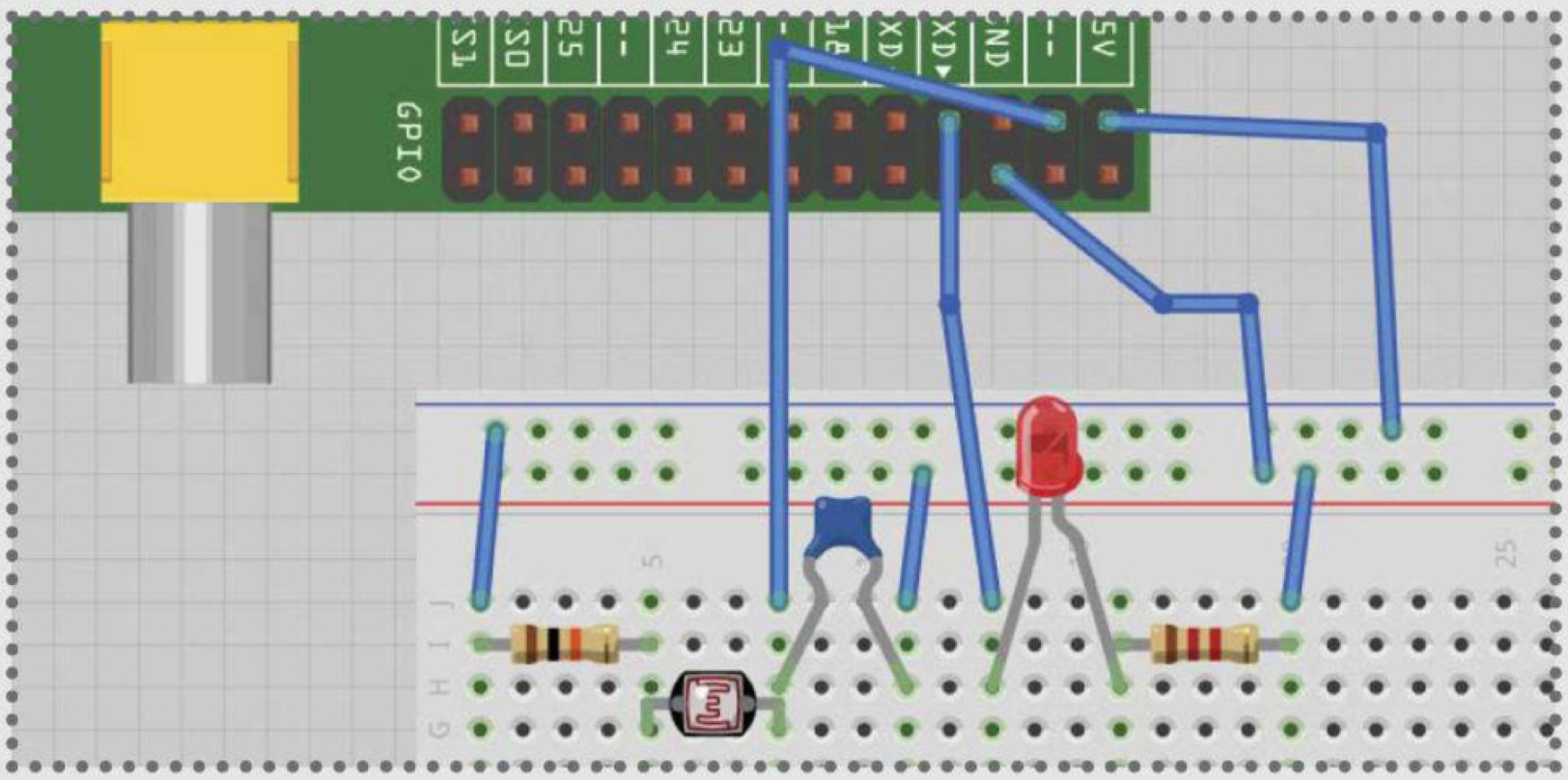


(LDR), or photoresistor, that changes its resistance based on the levels of light it's receiving.

There's no extra software you'll need for this, so just do an **apt-get update** and **apt-get upgrade**. Turn it off, unplug it from the power source and wire up the circuit as shown below. Note the placement of the components, especially the LDR and the capacitor. You'll need to measure the time it takes for the capacitor to fully charge between pin three of the GPIO ports and ground. Also, make sure the negative end of the capacitor is hooked to the ground side of the circuit. Next, turn your Pi on and download the code:

```
$ wget http://www.linuxuser.co.uk/wp-content/
uploads/2014/08/dusklight.zip
```

Unzip it and run it. We've made it so the Python shell will print out the values that the circuit returns, with a higher number meaning it's darker and the capacitor takes longer to charge. To get this working for outdoor use it will require a bit of trial and error. The easiest way is to set it up when it's getting to about the level of darkness you'd want the light to turn on and record the output from the sensor. Change the if statement so that it activates the LED over a certain number and you'll be set.



Left It's important to get the LDR and capacitor into the right positions

Put 'em together

Linking up the AirPi with your dusklights allows you to have the ultimate outdoor Raspberry Pi device, allowing you to predict the weather and control your outdoor lighting with more precision. The humidity and pressure sensors will be able to tell if it's foggy or not, giving you more control over when your lights turn on.



Outdoor time-lapse camera

Get beautiful views of the sunset using your Raspberry Pi, a Pi camera and a small Python script

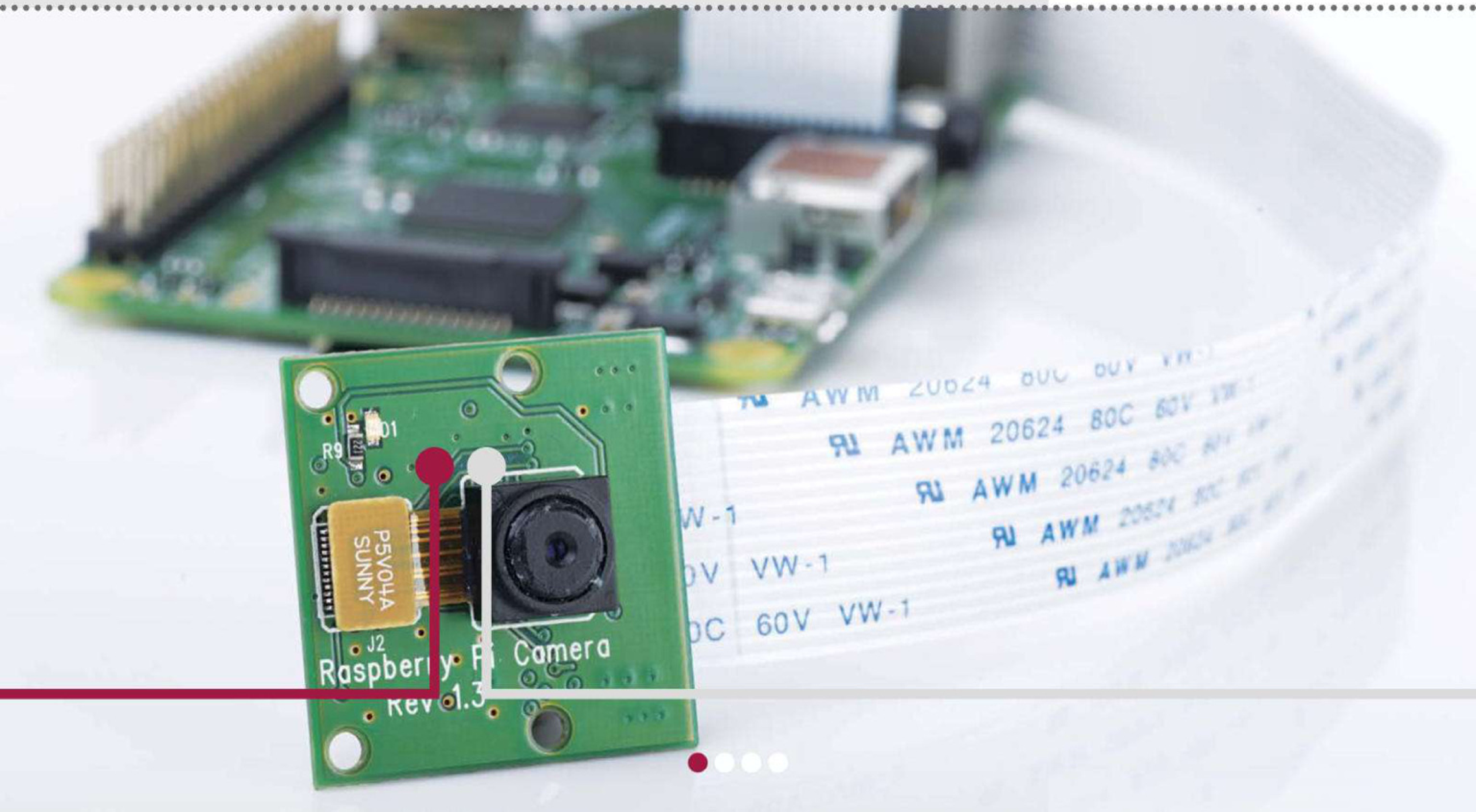
We love the Raspberry Pi camera. It's a lovely little piece of kit that is as versatile as the Pi and it doesn't even take up any of the USB slots. We've done a bit of time-lapse photography in the past but that was using a proper camera attached to the Pi – now we're doing it with just the Pi camera and a lot less code thanks to the picamera Python module.

We need to set up our Pi with a few things before we start with the code, though. We'll start off with assuming you've got a freshly installed Raspbian SD card, so the first thing to do is an **apt-get update** followed by an **apt-get**

THE PROJECT ESSENTIALS

Pi camera module

A weatherproof case, such as a PICE+

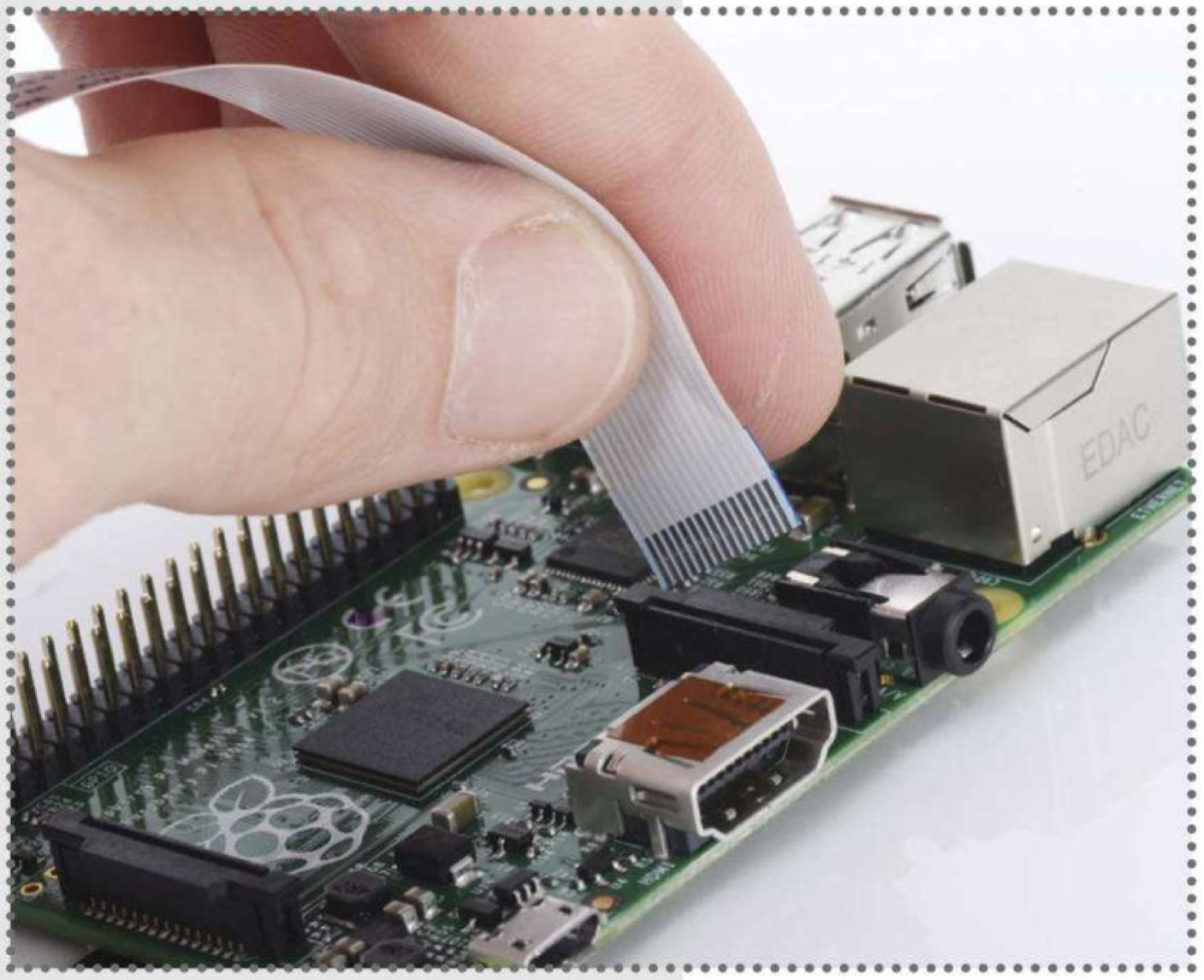


Below As well as the classic Pi camera module, you can get a NoIR version without an infrared filter, for night-time shoots

upgrade to make sure your files are up to date. Follow this with an **rpi-update** to make sure your firmware is also up to date – this step is very important because if you're using an outdated firmware then the camera won't work.

The next step is to get the camera enabled. Open up the terminal and enter:

```
$ sudo raspi-config
```



Above Ground yourself and carefully insert your Pi camera while the Raspberry Pi is unplugged

Find the Enable Camera option. Press Enter and key over to Enable. Go to Finish but don't restart straight away, as you have some more commands to get through. Install your picamera module using this:

```
$ sudo apt-get install python-picamera
```

You also need to install the software that you're going to use for compiling your images into the finished time-lapse video. We're using gstreamer to compile the video as it works well with the Raspberry Pi GPU. To get it installed, first add the repo to the sources list with:

```
$ sudo sh -c 'echo deb http://vontaene.de/
raspbian-updates/ . main >> /etc/apt/sources.list'
```

Do an **apt-get update** and then install a huge selection of packages with the following:

"We're using gstreamer to compile the video as it works well with the Raspberry Pi GPU"



Left The PICE+ case is currently being Kickstarted, so it wasn't available at the time of writing, but look out for it by the end of the year

```
$ sudo apt-get install libgstreamer1.0-0
liborc-0.4-0 gir1.2-gst-plugins-base-1.0 gir1.2-
gstreamer-1.0 gstreamer1.0-alsa gstreamer1.0-
omx gstreamer1.0-plugins-bad gstreamer1.0-
plugins-base gstreamer1.0-plugins-base-apps
gstreamer1.0-plugins-good gstreamer1.0-plugins-
ugly gstreamer1.0-pulseaudio gstreamer1.0-
tools gstreamer1.0-x libgstreamer-plugins-
bad1.0-0 libgstreamer-plugins-base1.0-0.
```

Once that's done, shut down the Raspberry Pi. Unplug the USB cable and locate the special DSI port for the camera, next to the HDMI port. Gently pull on the edges to lift the fastener and slot in your Raspberry Pi camera ribbon – make sure the silver connectors are facing the HDMI port. Turn your Raspberry Pi back on and get back into Raspbian. Everything should be about ready now, so give the camera a test by opening up the terminal and using:

```
$ raspistill -o test.jpeg
```

It will show the preview screen and then take a photo after five seconds; if it does so, you're ready for our code.

"Gently pull on the edges to lift the fastener and slot in your Raspberry Pi camera ribbon – make sure the silver connectors are facing the HDMI port"

Storemags.com

Our preferred method for doing this is to first create a folder called timelapse in your home directory:

```
$ mkdir timelapse
```

If you haven't made any changes to your username and such, this should create the directory with the full path /home/pi/timelapse. Keep this in mind and we'll explain its importance in a bit. Now in the terminal we'll download the Python code we've created for this project:

```
$ wget http://www.linuxuser.co.uk/wp-content/
uploads/2014/08/timelapse.zip
```

Unzip it and have a quick look through it using IDLE. There are a couple of things to note in the code, in case you wish to modify it. The photos variable is set to 500 and that's the number of shots it will take before compiling the video. In reality, it will actually do one more due to the mathematical quirks of Python, but that shouldn't matter. The delay parameter is in seconds and, using a bit of arithmetic, you can figure out that 500 shots at 30-second intervals will take about four hours to complete.

The large subprocess.call line is used to compile the images into an AVI file at 24 frames per second. It uses the full path to the directory that we mentioned previously, so if yours is slightly different make sure you change it now.

There are a few ways you can get a time-lapse of the outdoors; the easiest and safest is to find a window with the view you wish to use. This may not always be easy, though, in which case we suggest trying out the PICE or PICE+ case. It's durable, waterproof and designed to house the Pi and camera. Just power it with a long cable and run the code, then retrieve the Pi and the AVI file later.

Be inspired

With some modifications to the code and the Pi, you can make it a lot more automated, running the Python script after boot and uploading it to the cloud storage we set up earlier. You can also use it as a CCTV camera thanks to an additional hood that comes with the PICE

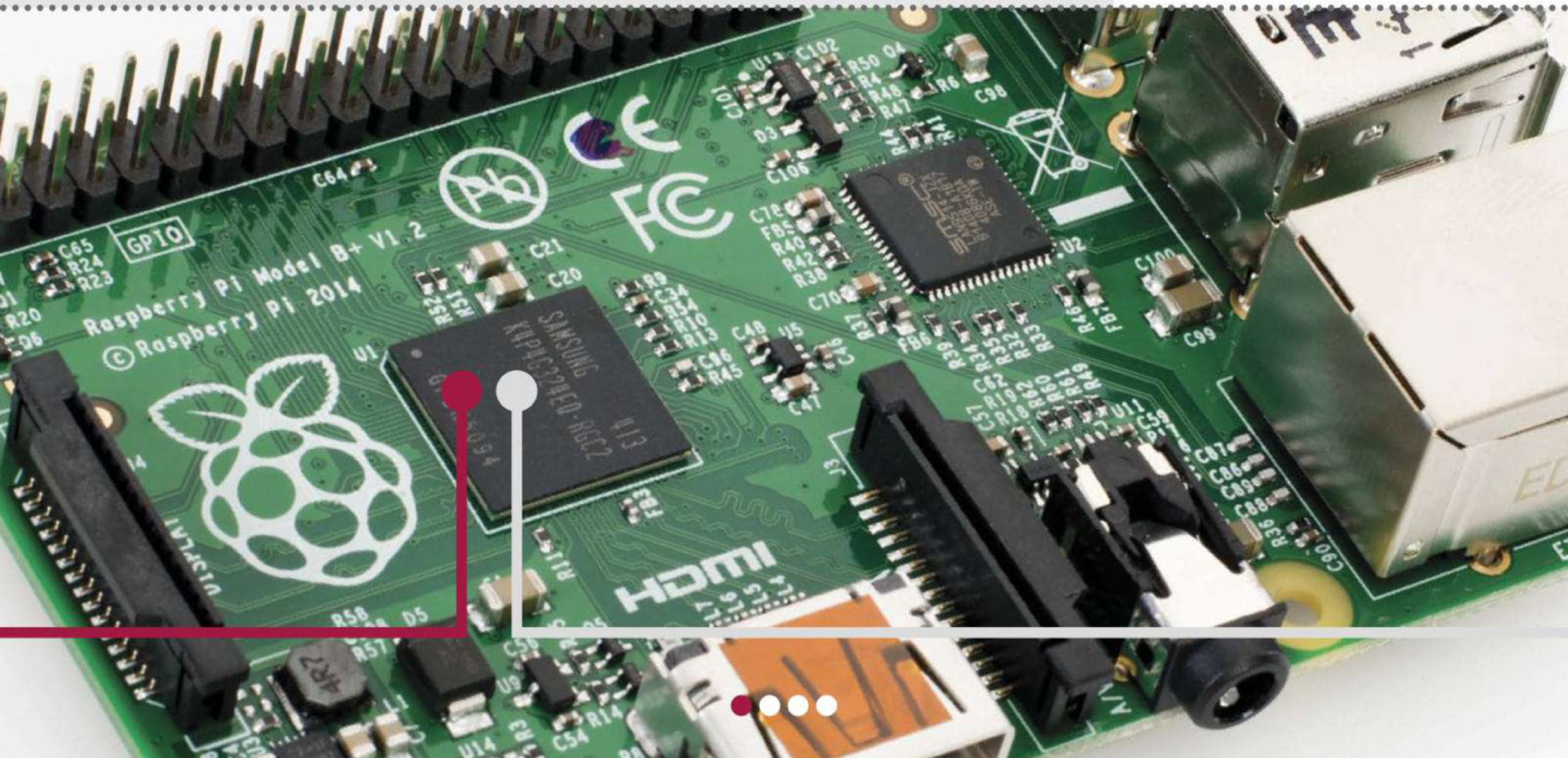


Further reading

Find even more inspiration inside these brilliant resources

If you've worked your way through all ten projects and even had a crack at combining some of them, you're likely wondering where to go next. Well, there are many places we can point you towards for even more inspiring projects, as well as places to shop for parts and read up on the technical details of anything you're not sure about. As well as the sources below, there's a wealth of Pi goodness to be found on our sister website **www.linuxuser.co.uk** – just search for 'Raspberry Pi'. We also recommend reading **Raspberry Pi: The Complete Manual** and **Understanding & Implementing Python**, which lives in the Special Editions slot of the Linux User & Developer app.

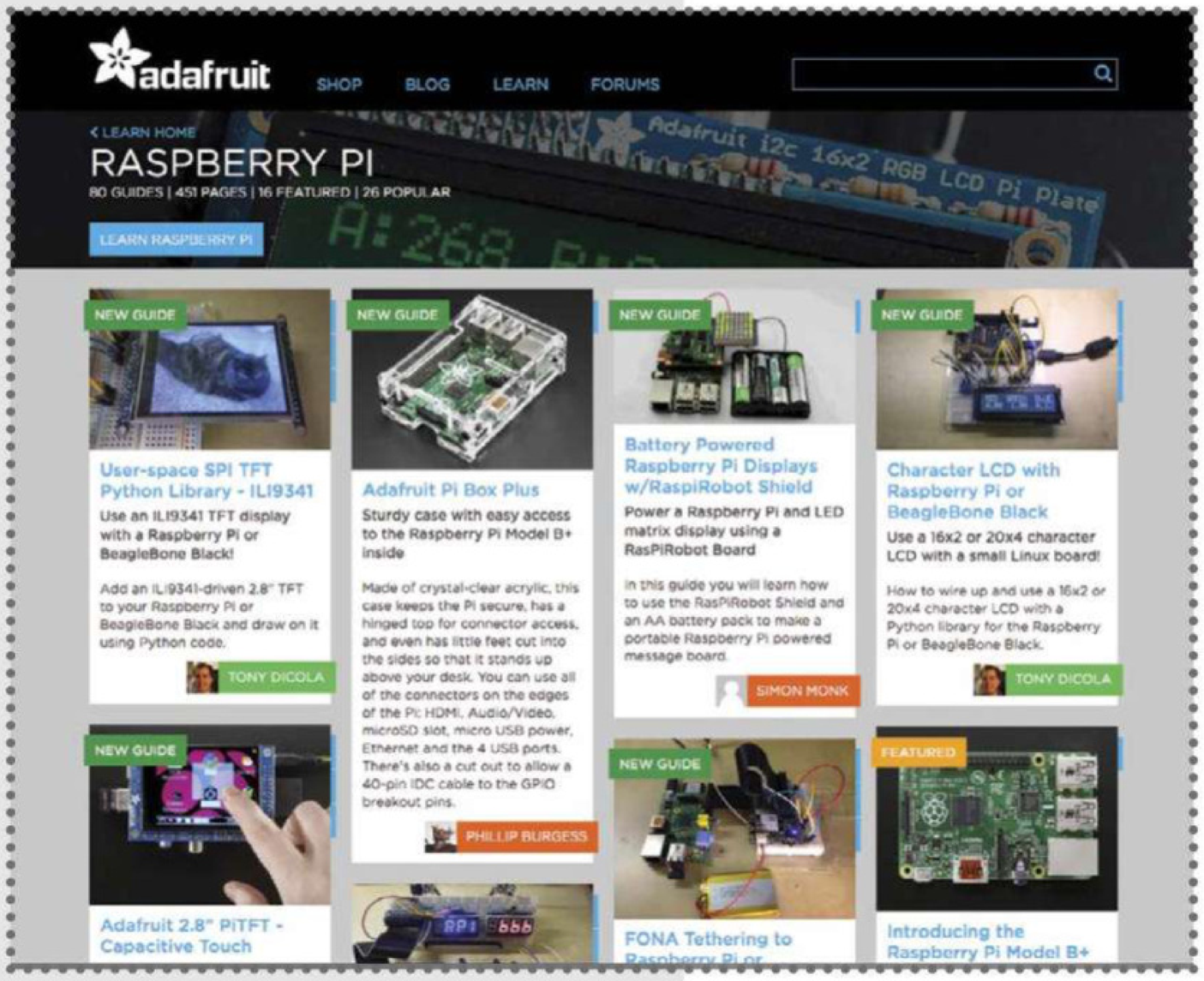
Below As well as the sources we recommend here, it's also worth heading over to a nearby Raspberry Pi Jam



Adafruit Learning System

If you're looking for more inspiration then the Adafruit Learning System is a brilliant place to begin, especially if you check out the main Raspberry Pi category. You can find dozens more projects here that will test your knowledge and help you push it even further, and if you find yourself needing any Adafruit modules or other parts then you can always grab them from the shop, which is linked right at the top of the page. It's a great little community to be a part of, and it's well worth creating an account and uploading your own projects too.

learn.adafruit.com/category/raspberry-pi



Above You can see the most recently uploaded projects in the Learning System

CPC

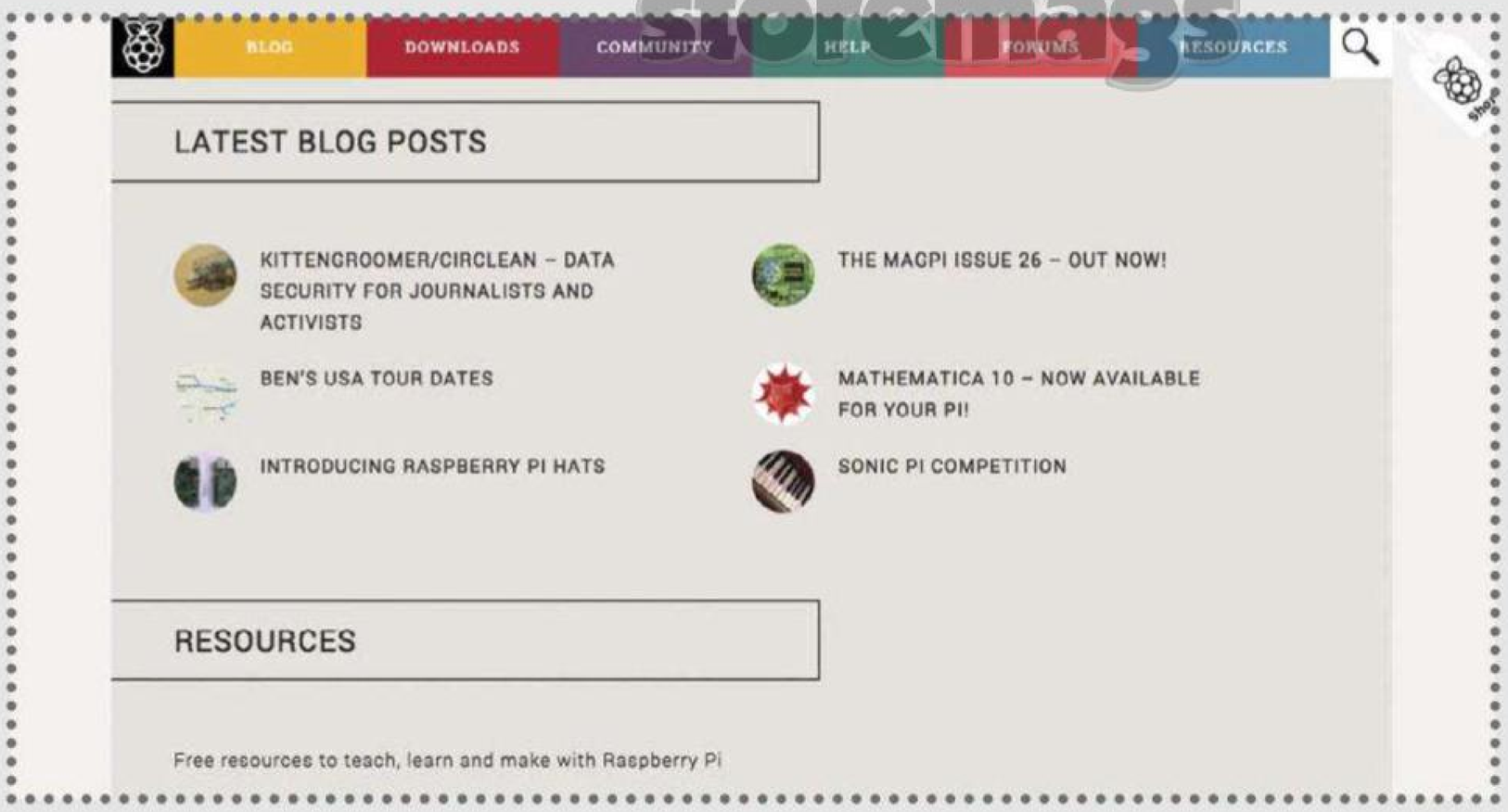
For electrical components, we recommend that you pay a visit to CPC. For this projects feature, we bought the majority of the components that we didn't already have lying around (like the time-lapse project's camera) from here. For UK orders placed online, you can even get free delivery and a same-day dispatch. CPC's catalogue really is comprehensive, so you can get pretty much any electrical component you need from them.

cpc.farnell.com

Raspberry Pi Foundation

As the popularity of the Raspberry Pi has grown, so has

"For electrical components, we recommend that you pay a visit to CPC"



the Foundation's website. New blog posts are going up all the time, announcing new kit like the Raspberry PI 2, and you'll find posts showing off some of the amazing work being done by the Raspberry Pi community as a whole. It's a great place to visit when you're looking for some ideas to spark off a new one of your own – there are a bunch of recommended sources to be found under the Community tab as well, so you can keep tabs on the latest projects.

www.raspberrypi.org/forums

Raspberry Pi Hub

All the technical information you could possibly need about the Raspberry Pi can be found over on the RPi Hub page of the eLinux wiki. There's a buyer's guide, a beginner's guide, resources for the hardware and software, links to the full documentation, and there are also round-ups of community projects and tutorials. If you head through to the Projects page then you'll find a massive list of projects and links to check out the source pages. So as well as being a good source of knowledge, the Raspberry Pi Hub can also help you find even more Pi projects to try.

elinux.org/RPi_Hub

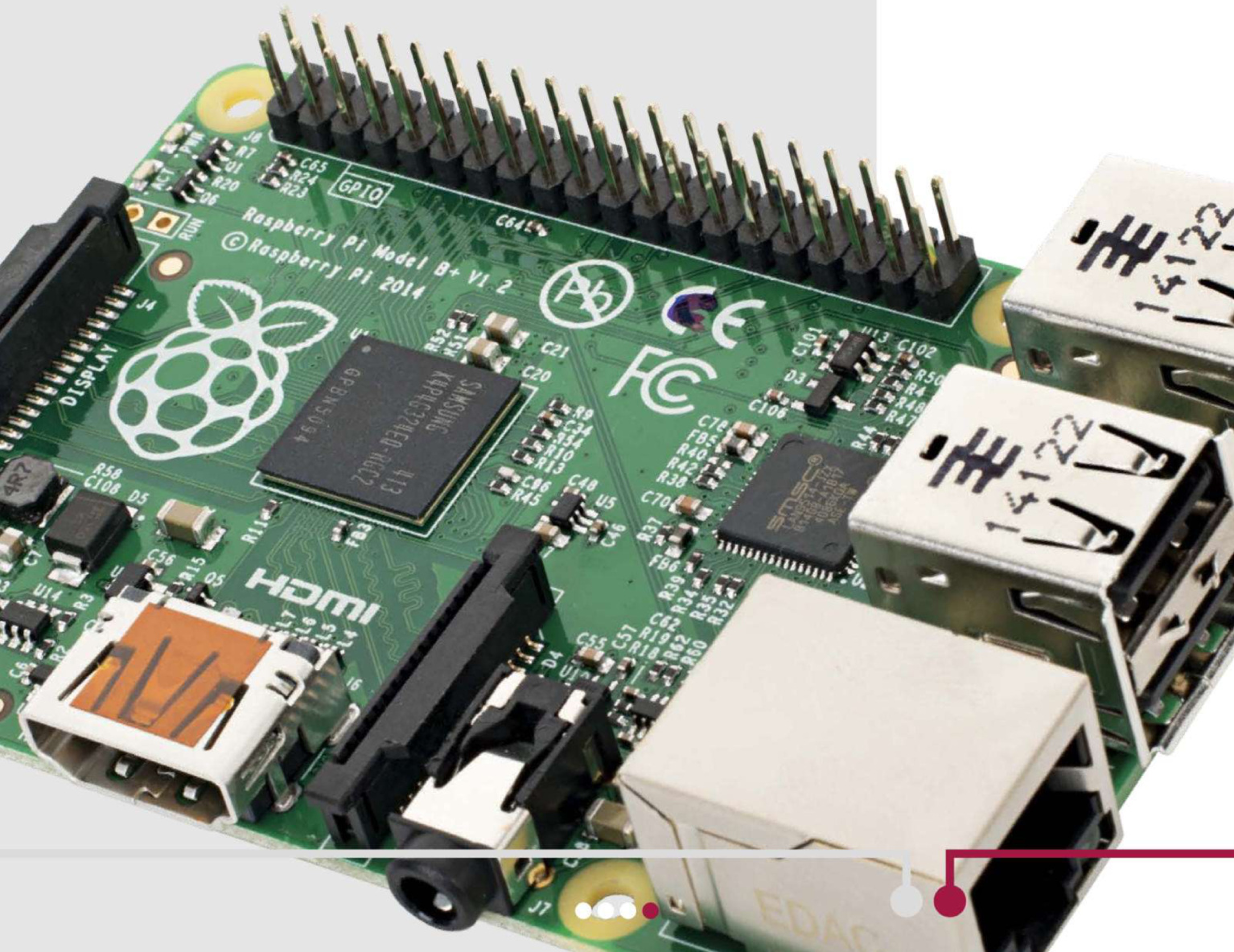
Left The Pi Foundation site really is lovely, and a lot of it is to do with the writing itself

"You'll find posts showing off some of the amazing work being done by the Raspberry Pi community. It's a great place to visit when you're looking for some ideas"

Raspberry Pi forum

There's more to the Raspberry Pi Foundation's website than you might think – head to the Forums link right at the top of the page and you'll find out where all the conversations are really happening. It's a really active community, with people are posting in questions, ideas and finished projects all the time, so it's a fantastic way to get involved by asking for or giving a hand with Pi projects. As with all resources like this, it's best to search through the forums before you do start asking questions, though, and to read the stickies, because people will likely have already asked the same thing already and you can just find that particular thread.

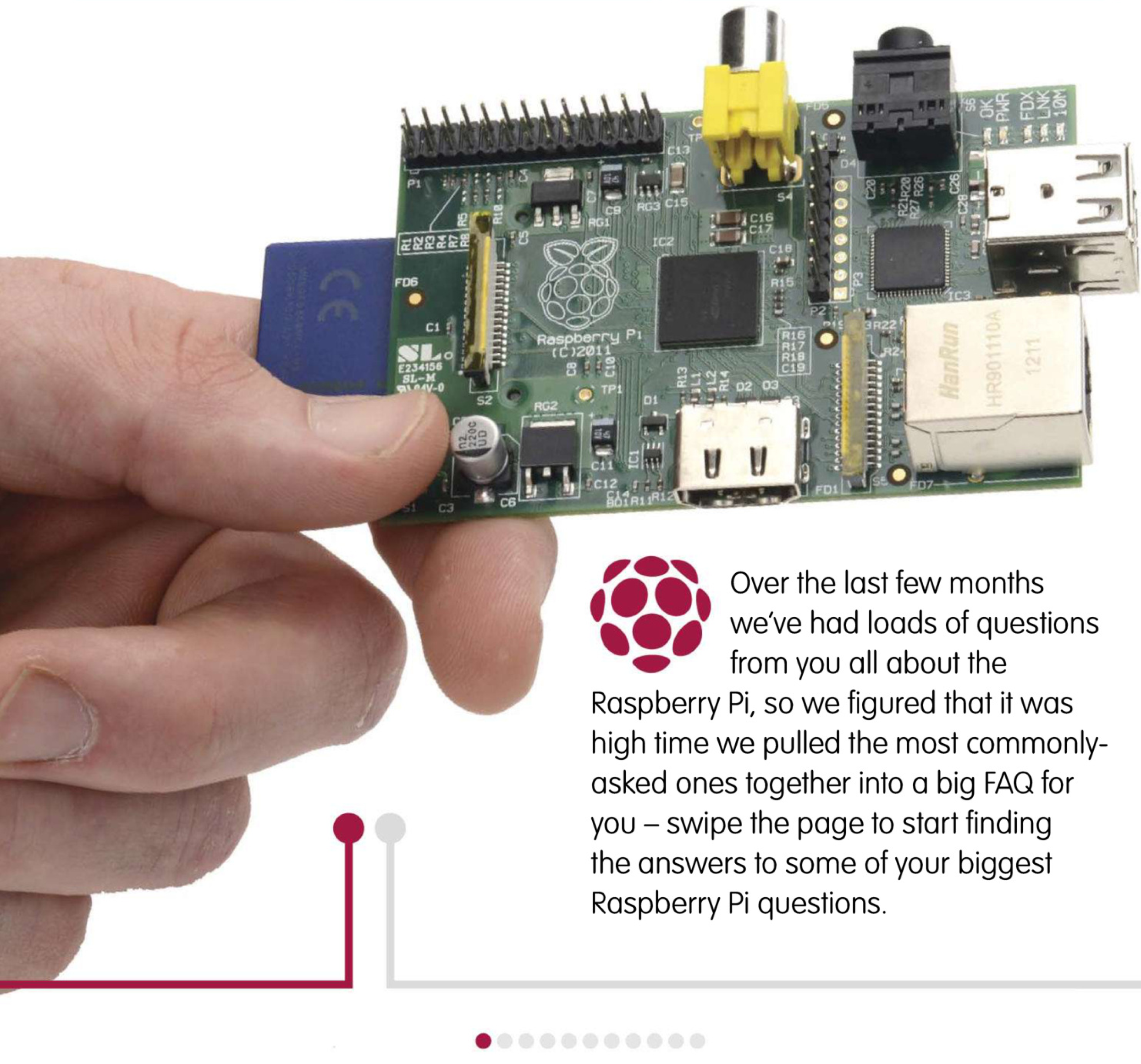
www.raspberrypi.org/forums



Talking Pi

Join the conversation at...

@linuxusermag | Linux User & Developer | RasPi@imagine-publishing.co.uk



Over the last few months we've had loads of questions from you all about the Raspberry Pi, so we figured that it was high time we pulled the most commonly-asked ones together into a big FAQ for you – swipe the page to start finding the answers to some of your biggest Raspberry Pi questions.

There are many questions thrown around about the Raspberry Pi, due to its wide range of both users and possible uses.

Its ease of accessibility and low cost make the Pi appealing to many different kinds of people, from complete computing novices to more advanced users who want to use it for out-of-the-ordinary projects because of the Pi's low-risk nature. This leads to a lot of different questions being asked about the Raspberry Pi: Why is it so cheap? What operating systems can I use on it? What SD cards can I use with it?

Well, we have put together a list of a few of the most commonly asked questions regarding the Raspberry Pi here. There's no way we could possibly cover everything that you've all asked in this one article, but hopefully it'll go some way towards explaining queries you have!

"The Raspberry Pi Foundation is a charitable organisation registered with the Charity Commission for England and Wales in May 2009"

Why is the Raspberry Pi so cheap?

The Raspberry Pi Foundation is a charitable organisation registered with the Charity Commission for England and Wales in May 2009. Its aim is to promote the study of computer science and related topics, especially at school level – and to put the fun back into learning computing.

The purpose of the Pi was to create a small computer that was accessible to every kind of end user – from the average home user that wants something to play around with or come up with a new and inventive use, to the education sector where, currently, technology in many parts of this country (and across the world) is inaccessible in schools due to cost, leading to little-to-no understanding of computer science and creating a big gap in the skills that children in schools are learning.

The price you pay reflects the cost of manufacturing and shipping costs for the assembled alone – nothing more, nothing less.

There were two versions of the Pi to begin with – the Model A and Model B boards. The main distinguishing factor between the two is that the Model B has 512MB RAM and an Ethernet port, whereas the Model A has 256MB RAM, no Ethernet port and just a single USB port.

A lot of effort and research went into the Raspberry Pi, keeping costs down to a minimum by removing certain components that weren't critical – such as an on-board real-time clock which required a battery and drove up prices – and made the physical requirement for space much larger. Lots of non-critical components can be added to the Pi should you need them.

Where can I get a good case for my Raspberry Pi?

It's not ideal using the Raspberry Pi on its own, out of a case. It can be quite vulnerable and easily breakable if it's not in some way enclosed or in a dedicated case.

There are a lot of people now selling cases for your new Raspberry Pi. The first commonly available case was the Raspberry Pi case from ModMyPi (**www.modmypi.com**) – it features a two-part moulded plastic case that can be ordered in two different colours.

The other common Raspberry Pi case is the 'BerryBlack' case – available from **http://shop.pimoroni.com**. This again features a two-part plastic case, but requires Model B users to knock out some plastic inserts to be usable.

Other than these, a quick Google or eBay search will reveal lots of different cases available.

It's worth noting that given the relative size and lack of complexity of the Raspberry Pi board, there's a vast

"The first commonly available case was the Raspberry Pi case from ModMyPi"

Storemags.com

community of people that create their own cases – in anything from Lego boxes to cigarette rolling tins. So long as you can get the required ports in the required places, you can use whatever you like.

In the Raspberry Pi site archives, there's also the option to print off a template for a DIY case – made entirely from cardboard. Not exactly sturdy, but a good way to make your Pi a little more noticeable. See **www.raspberrypi.org/archives/1310**.



Above A Raspberry Pi sporting a ModMyPi case. You don't have to buy one, though – why not try 3D-printing your own?

What are my options for audio on the Raspberry Pi?

The Pi is able to play high-quality audio. It has two options: a 3.5mm audio jack, and audio out over HDMI.

These open up options of getting audio out through almost anything. A set of PC speakers will plug into the Pi's 3.5mm audio jack, but this is also what you'd use to plug in your headphones.

At a slightly higher level, rather than using the analogue audio jack you could use the HDMI port, which means that the audio goes out digitally. As well as giving a higher quality of audio, it also means that you can run this into your TV or other HDMI advice, then out through your TV's optical connection, which may go to your surround sound system or a whole bunch of other stuff.

If you want to specifically switch between the two audio outputs, you can do so at the ALSA mixer level by using the following command in your terminal:

"Rather than using the analogue audio jack you could use the HDMI port, which means that the audio goes out digitally"



Left The cable at the bottom is a standard 3.5mm audio jack, and above it is an HDMI cable. You can use either to output Raspberry Pi audio

```
amixer cset numid=3 <n>
```

Where n is:
0 – auto
1 – audio jack
2 – HDMI

What's the biggest SD card the Raspberry Pi supports?

Officially, the biggest SD card size that's been tested by the Raspberry Pi team is a 32GB card.

As far as choosing an SD card goes, the best way to select one is just to not go for the absolute cheapest one you can find. Instead, go with something of a recognised brand. While class isn't critical, it's worth noting that the Raspberry Pi will be considerably slower with a card of a lower class.

There is a list of officially tested cards on the eLinux site – using this you'll be able to look for cards of a specific size and/or class that meet your requirements: **http://elinux.org/RPi_SD_cards**.

On here there's only one tested 32GB card, but it's confirmed to work without issue. The minimum sized card

Below This is an SD card – it's your Raspberry Pi's boot and storage device, though you can add more with a USB drive



storemags.com

you can use is 2GB but it is recommended to use at least 4GB to run additional programs on it.

If you simply require more storage space for your Raspberry Pi, one of the easiest and cheapest ways to achieve this is to have a smaller SD card just to run your operating system, a USB hub and a powered USB drive to store all your data. Mechanical hard drives are still much cheaper to buy than flash storage, but the drive would be bigger than the Pi – and as the Pi has limited power available, it requires powering separately.

Can I overclock my Raspberry Pi for more speed?

Yes you can, but if you start seeing problems it's best to switch back to your original settings – or keep dropping them until you maintain stability.

You can set your overclock settings with the raspi-config tool, or by modifying the /boot/config.txt file on your SD card instead.

There are several different presets for overclocking in the raspi-config tool. Generally speaking, the 'Modest' overclock gives a little performance increase all round, and is the safest and most stable for the hardware.

If you'd like to be a bit more adventurous and play around with the config.txt for overclocking, you should just be able to modify the values:

```
arm_freq=xxx
sdram_freq=xxx
```

Beware that this is clearly more dangerous, since you're inputting raw values that are to be read upon startup of the system.

It's also worth noting that if you manually modify the config.txt and set:

```
force_turbo=1
```

… your warranty will be null and void.

"The 'Modest' overclock gives a little performance increase all round"

Storemags

It's also possible to check the current CPU clock speed using the terminal and a simple command:

```
cat /proc/cpuinfo
```

You'll see a few different lines of information being thrown out, but the one you're most interested in is the second which reads 'BogoMIPS' – by default, this should be somewhere around 700.

Each time you make changes to your Raspberry Pi's overclocking settings and have rebooted the system, run this again and you should see that this number has changed in accordance with what you've set as your desired clock speed.

Can I add some more memory to my Raspberry Pi?

Unfortunately not. The Raspberry Pi is a single PCB. The memory is not removable – nor is it swappable. If you buy a 256MB Model A, you're stuck with a 256MB Model A. You are unable to use any means to upgrade your Raspberry Pi, be it memory, CPU or graphics.

You are able to overclock your device, though (see above). This will potentially allow all components to run marginally faster, but you cannot add more of them.

How can I connect more devices to my Raspberry Pi?

The easiest way to be able to connect more devices to your Raspberry Pi is by using a USB 2.0 hub. There are two kinds of USB hubs – powered and non-powered.

If you just want to connect very low-powered devices such as a USB flash drive, keyboard and mouse, you should be okay with a non-powered hub.

However, if you want to add other devices – such as a USB-powered hard disk – then you'll require a powered

"You are unable to use any means to upgrade your Raspberry Pi, be it memory, CPU or graphics. You are able to overclock your device, though"

StoreMags



Left This is a powered USB hub. It is known as 'powered' because it has a power supply to provide power to connected devices

hub (one that comes with its own power supply.) Failure to use a hub of this kind will often lead to your devices not being recognised by the Raspberry Pi and being left inaccessible – or, in the worst case scenario, leading to system instability.

If you're making a decision about a hub and don't know what to get, always go for one that's powered.

What operating systems can I run on my Pi?

Already, there are a lot of different operating systems available for the Raspberry Pi. Anything that runs on an ARM system should be okay.

However, there are a few distributions that are favoured in the community.

Debian (Squeeze) was the default distro used on the early versions of the Raspberry Pi board. It was the first fully functional OS with a desktop, browser and set of development tools.

Raspbian is the most common all-rounder and is used as a generic, all-purpose operating system. It's based on the Debian distro.

"If you're making a decision about a hub and don't know what to get, always go for one that's powered"

storemags

Arch Linux ARM is the ARM variant of Arch Linux. It's another general-purpose operating system, specifically lightweight (which means small) and simple, but not so easy to use for novices.

RISC OS is another desktop-driven OS, originally used on Acorn computers. Stable releases are not that common, though there are a few more unstable releases and continuing development.

OpenELEC and Raspbmc are both standalone operating systems that run Kodi – a common media-centre application for playing music/video and looking at pictures. If you run either OpenELEC or Raspbmc, it will boot straight into Kodi – there is no other user interface. They are single-purpose distros.

Is it possible to boot from a USB device?

No – the config.txt is the first thing that's read upon starting the Raspberry Pi. Before the CPU's initialised, the GPU checks this file on the SD card, at which point there's no USB support.

For this reason, it is unfortunately not possible to start a Raspberry Pi without an SD card being present with some sort of data on it.

It is, however, possible to have a USB boot device take over duties as the main storage device once the system has booted.

What programming languages can I use on the Pi?

The 'official' programming language that's being

"It's not possible to start a Raspberry Pi without an SD card being present"



Left It's not possible to boot directly from USB, although it can be used after boot for more storage

supported as the educational learning language on the Raspberry Pi is Python. However, any programming language that will compile for ARMv6, or ARMv7 if you have a Raspberry Pi 2, is usable as a means for making applications for the Pi.

Two languages have been included by default in Raspbian. These are:

Scratch – This is an entry-level language that doesn't require syntactically correct code, but helps you understand mathematical and computational concepts.

Python – As it's the 'official' language of the Pi, it'd be silly not to include it, right? It's very powerful and not too difficult to learn.

Other languages that'll work on the Pi already include several other widely known, lower-level ones: HTML5, Java, JavaScript, C, C++ and Perl.

To use and compile any of these directly on the Pi, you'll need to install your own compilers – though these can be obtained using the Aptitude package manager without too much hassle.

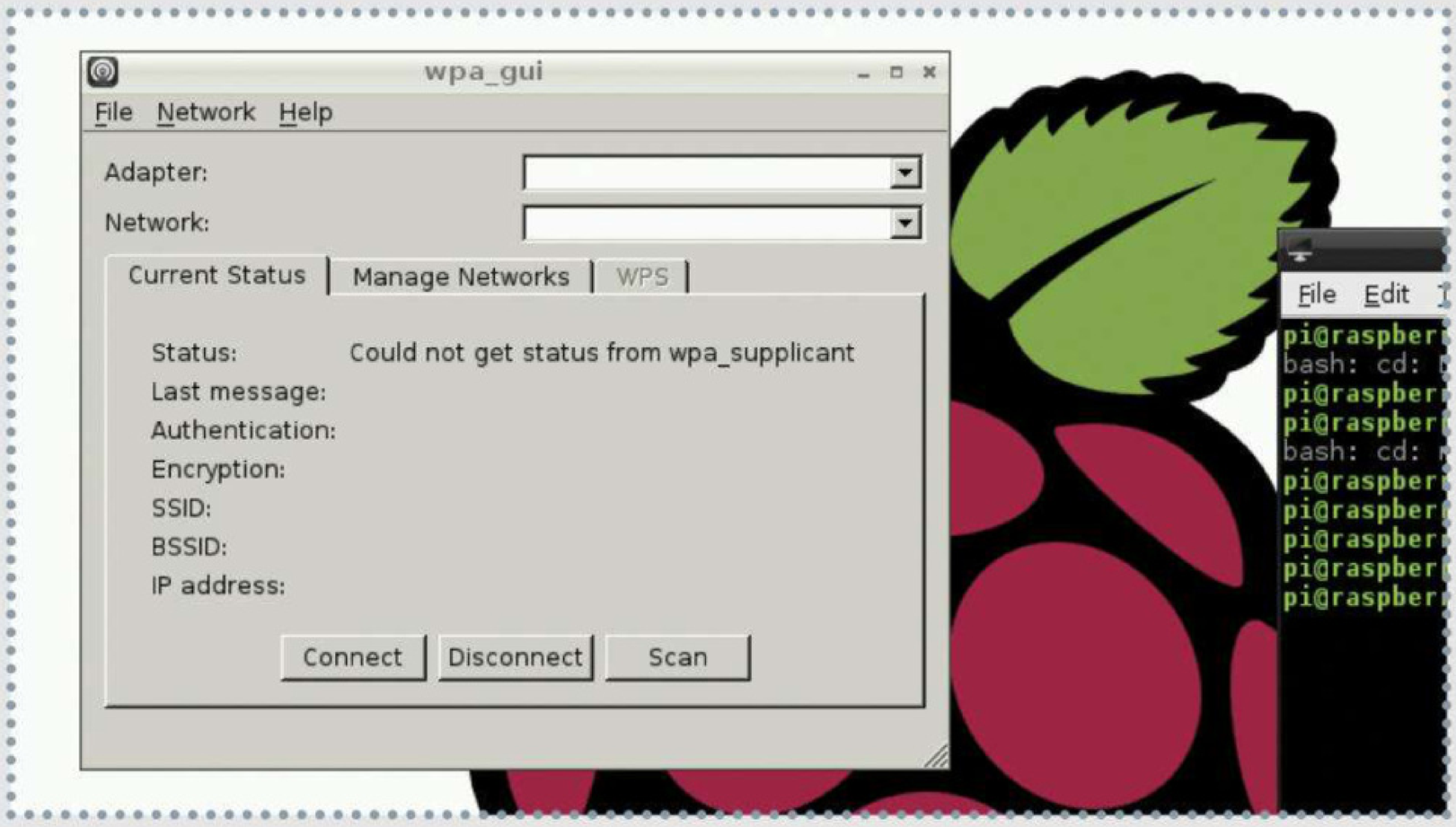
> "Other languages that'll work on the Pi already include HTML5, Java, JavaScript, C, C++ and Perl"

How can I add Internet to my Raspberry Pi?

01 By ethernet cable

Generally, the least configuration-intensive way is to use a network cable. It guarantees compatibility because it'll work out of the box with any of the available Pi distros. Simply plug both ends of the cable in and let the Pi do the rest; it should go and get an IP address.





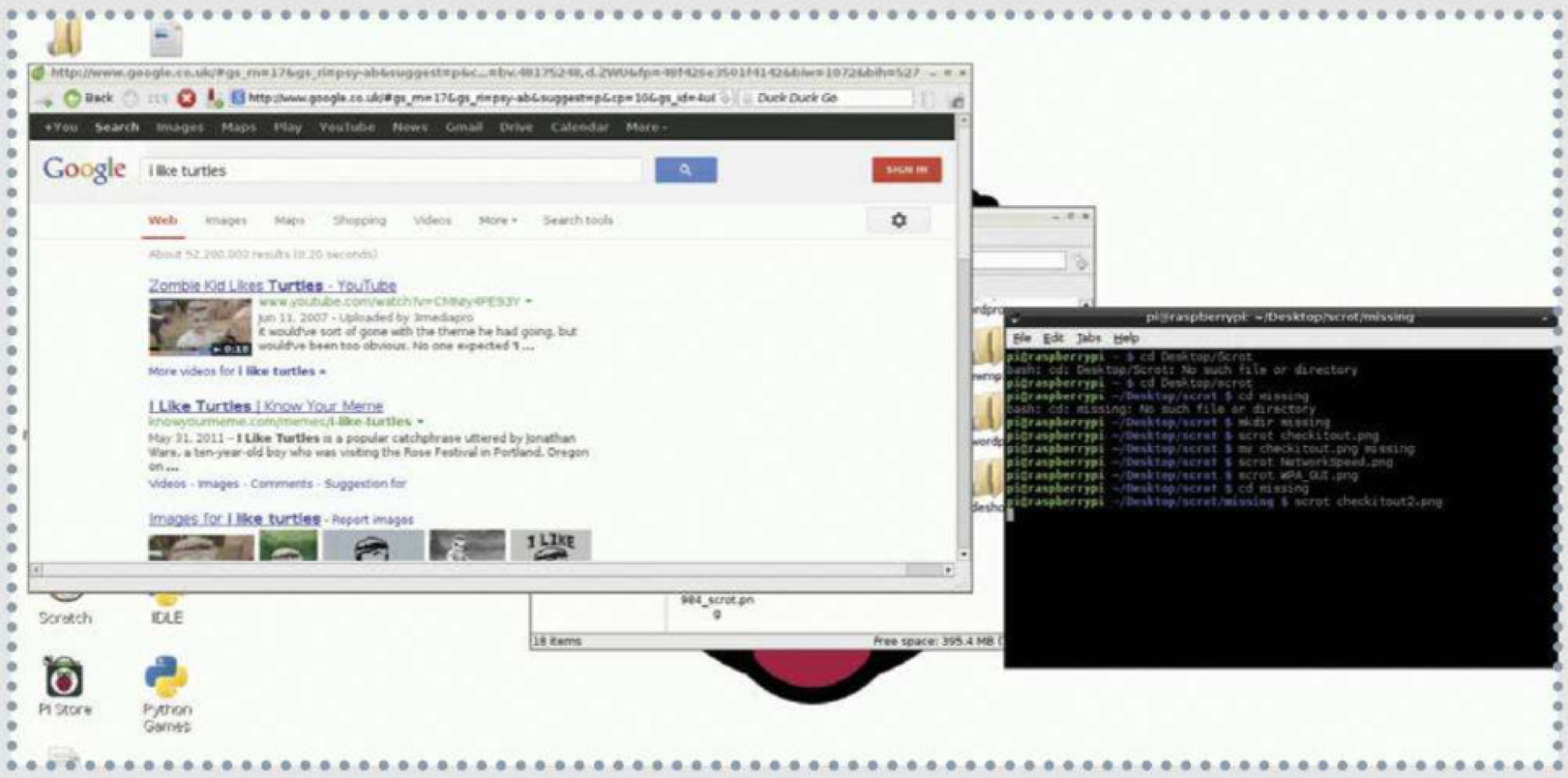
02 By wireless dongle

Plug your Wi-Fi dongle in and if using Raspbian, use the 'WiFi config' tool that lives on the desktop. Select your adaptor and network, then type in your network key. As long as your dongle is supported by the Pi, this should be all you need to get going.

"Use the 'WiFi config' tool that lives on the desktop. Select your adaptor and network, then type in your network key"

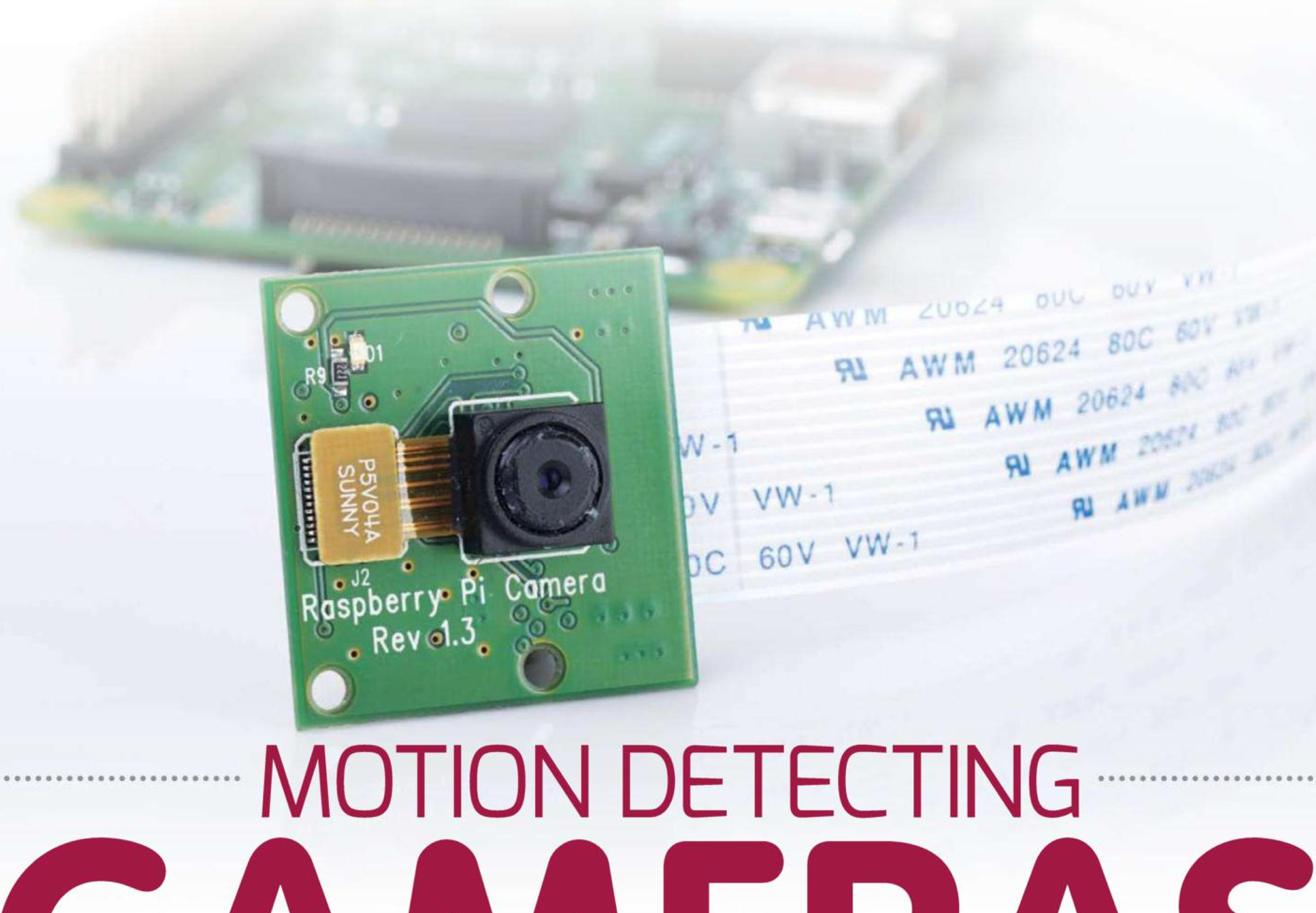
03 Check it out!

When you've done either of the steps above, start your favourite browser and do the age-old 'am I connected?' test by trying to search for something online. If you yield any results, you can be sure that your connection is working. If you get any errors, things might not have gone so well.



Next issue

Get inspired **Expert advice** **Easy-to-follow guides**



MOTION DETECTING CAMERAS

Get this issue's source code at:
www.linuxuser.co.uk/raspicode